



**PROTOTIPE PENGENALAN ABJAD JARI UNTUK TUNA RUNGU BERBASIS
MIKROKONTROLER ATMEGA 32**

PROYEK AKHIR

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta untuk
Memenuhi Sebagian Persyaratan Guna Memperoleh
Gelar Ahli Madya Teknik



OLEH:

**AAN SETIAWAN
NIM. 09507131013**

**PROGRAM STUDI TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2013**

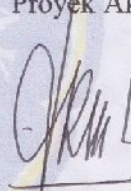
PERSETUJUAN

Proyek akhir yang berjudul **PROTOTIPE PENGENALAN ABJAD JARI
UNTUK TUNA RUNGU BERBASIS MIKROKONTROLER ATMEGA 32 "**
ini telah disetujui oleh pembimbing untuk diujikan.

Yogyakarta, 18 Maret 2013

Menyetujui,

Dosen Pembimbing
Proyek Akhir



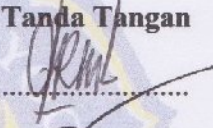
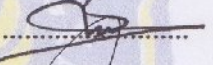
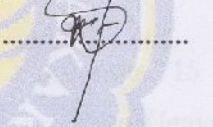
Aris Nasuha, S.Si, MT
NIP. 19690615 199403 1 002



PENGESAHAN

Proyek akhir yang berjudul "PROTOTIPE PENGENALAN ABJAD JARI
UNTUK TUNA RUNGU BERBASIS MIKROKONTROLER ATMEGA 32"
ini telah dipertahankan di depan Dewan Penguji pada tanggal 1 April 2013
dan dinyatakan lulus.

Dewan Penguji

Nama	Jabatan	Tanda Tangan	Tanggal
Aris Nasuha, S.Si, MT	Ketua Penguji		15-09-2013
Drs. Djoko Santoso, M.Pd	Sekretaris Penguji		09-09-2013
Drs. Suparman, M.Pd	Penguji		09-09-2013

Yogyakarta, 15 April 2012

Fakultas Teknik

Universitas Negeri Yogyakarta

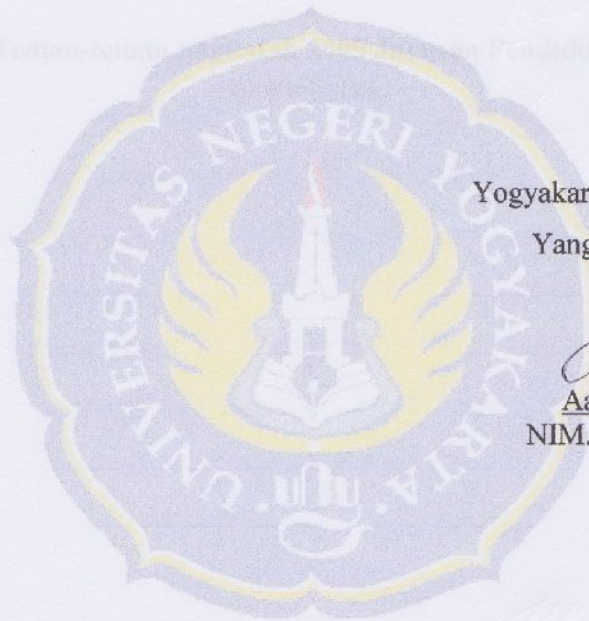
Dekan,



Dr. Moch. Bruri Triyono
NIP. 19560216 198603 1 003

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Proyek Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya atau gelar lainnya di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.



Yogyakarta, 15 Maret 2013

Yang Menyatakan,

Aan Setiawan

NIM. 09507131013

PERSEMBAHAN

Dengan rasa syukur yang mendalam Tugas Akhir ini ku persembahkan kepada :

Bapak dan Ibu tercinta yang telah memberikan semangat dan doa serta fasilitas dalam menyelesaikan Tugas Akhir.

Adik - adik ku dan Keluarga Besar ku Terimakasih segalanya.

Seluruh Teman-teman angkatan 2009 Jurusan Pendidikan Teknik Elektronika.

MOTTO

Yang pertama belum tentu yang TERBAIK, tetapi bisa jadi yang terbaik adalah yang TERAKHIR.

BERUSAHALAH MENJADI LEBIH BAIK DAN TERBAIK

Kegagalan dan keberhasilan bukanlah takdir namun sebuah pilihan.

Berfikir sejenak, merenung masa lalu adalah permulaan yang baik untuk bertindak

Berusahalah untuk menjadi yang terbaik, tetapi jangan pernah merasa bahwa kita yang terbaik

PROYEK AKHIR

PROTOTYPE PENGENALAN ABJAD JARI UNTUK TUNA RUNGU BERBASIS MIKROKONTROLER ATMEGA 32

Oleh : Aan Setiawan

NIM : 09507131013

ABSTRAK

Tujuan pembuatan prototipe pengenalan abjad jari untuk tuna rungu berbasis ATmega 32 adalah untuk membuat suatu alat yang bisa membantu tuna rungu dalam mempelajari huruf abjad dengan menggunakan tampilan LCD grafik sebagai penampil gambar isyarat jari. Pembuatan alat ini juga untuk mengetahui unjuk kerja dari komponen – komponen yang digunakan sebagai penyusun utama alat ini.

Prototipe pengenalan abjad jari untuk tuna rungu berbasis ATmega 32 yang dirancang khusus untuk membantu tuna rungu dalam mempelajari huruf abjad dengan menggunakan tampilan LCD grafik sebagai penampil gambar isyarat jari. Alat ini akan bekerja ketika sistem dihidupkan. Kemudian LCD grafik akan menampilkan tampilan awal. Selanjutnya instruksi terfokus pada keypad matriks sebagai input huruf abjad. Instruksi berupa kode keypad dengan fungsi setiap tombol. Selanjutnya LCD grafik akan menampilkan karakter gambar isyarat jari dan karakter huruf abjad jari sesuai dengan penekanan tombol. Metode yang digunakan dalam pembuatan Prototipe pengenalan abjad jari untuk tuna rungu berbasis ATmega 32 berbasis mikrokontroler ATmega 32 ini adalah eksperimental. Metode ini terdiri dari beberapa tahap yaitu: (1) Identifikasi kebutuhan, (2) Analisis Kebutuhan, (3) Perancangan perangkat keras dan perangkat lunak, (4) Pembuatan alat, (5) Pengujian Alat, (6) Pengoperasian Alat dan (7) Masukkan pakar. Perangkat keras terdiri dari (1) Sistem minimum ATmega 32 sebagai pengendali utama, (2) Keypad matriks sebagai input huruf abjad, (3) LCD grafik sebagai penampil karakter gambar dan huruf abjad jari dan (4) Tombol *on/off* untuk menghidupkan atau mematikan sistem. Perangkat lunak terdiri dari (1) Definisi *prosesor*, (2) Penyertaan fungsi, (3) Definisi *Port*, Deklarasi variabel dan (4) Fungsi Utama.

Berdasarkan hasil pengujian yang telah dilaksanakan maka dapat disimpulkan bahwa prototipe ini dapat bekerja dengan baik. Prototipe pengenalan abjad jari ini dapat menampilkan 26 karakter gambar abjad jari dengan keterangan huruf abjad pada keypad. Selain itu terdapat tombol khusus untuk mengetahui cara penggunaan alat atau petunjuk pemakaian alat.

Kata Kunci :Keypad Matriks, ATmega32, LCD Grafik

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT atas berkat dan rahmat-Nya, sehingga penulis mampu menyelesaikan laporan proyek akhir yang berjudul “Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu Berbasis Mikrokontroler ATmega32”. Tujuan dari penyusunan Proyek Akhir ini adalah sebagai syarat kelulusan pada program studi Teknik Elektronika D3 Universitas Negeri Yogyakarta.

Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan laporan Tugas Akhir ini tidak akan lancar. Oleh karena itu pada kesempatan ini, izinkanlah penulis menyampaikan ucapan terima kasih kepada:

1. Dr. Moch. Bruri Triyono, selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
2. Drs. Muhammad Munir, M.Pd selaku Ketua Jurusan Pendidikan Teknik Elektronika.
3. Drs. Djoko Santoso, M.Pd. selaku Ketua Program Studi Teknik Elektronika dan Koordinator Proyek Akhir.
4. Aris Nasuha, S.Si, MT, selaku Pembimbing Proyek Akhir.
5. Seluruh Dosen dan Karyawan di Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta. Yang telah mendidik dan memotivasi selama kuliah di UNY.
6. Keluarga besar yang telah memberikan kasih sayang dan motivasi selama ini .
7. Adik penulis, Irfan Nurdiawan sebagai inspirasi dalam proyek akhir ini.
8. Rian, Irwan, Taufiq, dan teman-teman lainnya terimakasih atas bantuanya.

9. Teman-teman mahasiswa Teknik Elektronika UNY angkatan 2009
10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dalam penyelesaian laporan ini.

Akhir kata penulis berharap semoga tugas akhir ini berguna bagi siapa saja yang membacanya khususnya bagi penulis sendiri serta dapat memenuhi sasaran dan tujuannya bagi kemajuan ilmu pengetahuan dan teknologi di Indonesia.

Yogyakarta, 14 Maret 2013

Aan Setiawan

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
ABSTRAK.....	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	3
C. Batasan Masalah	3
D. Rumusan Masalah	4
E. Tujuan	4
F. Manfaat	5
G. Keaslian Gagasan	5

BAB II PENDEKATAN PEMECAHAN MASALAH

A. Tuna Rungu	7
B. Metode Abjad Jari	7
C. Perangkat Keras (<i>hardware</i>)	
1. Keypad Matriks	8
2. LCD Grafik 128 x 64	10
3. Mikrokontroler AVR	16
a. Arsitektur ATmega 32.....	16
b. Fitur ATmega 32	16
c. Konfigurasi Pin ATmega 32.....	18
d. Peta Memori.....	21
D. Perangkat Lunak (<i>software</i>)	23
1. Bascom Avr	24

BAB III KONSEP RANCANGAN

A. Identifikasi Kebutuhan	27
B. Analisis Kebutuhan	27
C. Perancangan Alat	28
1. Perancangan Perangkat Keras (<i>Hardware</i>)	28
2. Perencanaan Rangkaian.....	30
3. Langkah Pembuatan Alat	32
4. Perancangan Perangkat Lunak.....	35
D. Pengoperasian Alat	42

BAB IV PENGUJIAN DAN PEMBAHASAN

A. Hasil Pengujian	43
1. Pengujian Catu Daya	43
2. Keypad Matriks 5 x 6	43
3. Pengujian LCD grafik	44
4. Pengujian Seluruh Sistem	45
B. Pembahasan	46
1. Perangkat Keras (hardware).....	46
a. Catu Daya	46
b. Keypad Matriks 5 x 6	47
c. LCD Grafik	47
2. Software.....	48
C. Unjuk Kerja	50
D. Pembahasan Masukan Pakar Tuna Rungu... ..	51

BAB V KESIMPULAN DAN SARAN

A. Kesimpulan	52
B. Saran	53

DAFTAR PUSTAKA	54
-----------------------------	----

LAMPIRAN	55
-----------------------	----

DAFTAR TABEL

Tabel 1. Fungsi masing – masing pin <i>graphic</i> LCD.....	12
Tabel 2. Tipe-tipe data	25
Tabel 3. Instruksi dasar Bascom AVR.....	26
Tabel 4. Daftar alat yang dibutuhkan.....	32
Tabel 5. Daftar komponen yang dibutuhkan.....	32
Tabel 6. Pengujian Input dan Output Rangkaian Regulator.....	43
Tabel 7. Pengujian penekanan tombol pada keypad Matriks 5x6	43
Tabel 8. Pengujian Sistem Secara Keseluruhan	45

DAFTAR GAMBAR

Gambar 1. Abjad Jari	8
Gambar 2. Skema Keypad Matrix	9
Gambar 3. Cara kerja Keypad Matriks	10
Gambar 4. Dimensi Graphic LCD WG12864C-TMI	11
Gambar 5. Pembagian IC KS0108B	13
Gambar 6. Blok Diagram <i>Graphic</i> LCD	14
Gambar 7. Urutan <i>Write Timing</i>	15
Gambar 8. Urutan <i>Read Timing</i>	15
Gambar 9. Blok Diagram AVR ATmega 32	17
Gambar 10. Konfigurasi Pin ATmega 32	18
Gambar 11. Peta Program memori	21
Gambar 12. Peta Data Memori	22
Gambar 13. Tampilan Jendela Program BASCOM AVR	24
Gambar 14. Diagram Blok Sistem Prototipe Alat	28
Gambar 15. Rangkaian sistem minimum ATmega 32	39
Gambar 16. Rangkaian catu daya	30
Gambar 17. Rangkaian Prototipe Pengenalan Abjad Jari	31
Gambar 18. Bagian Box	33
Gambar 19. Pengaturan Chip Pada Bascom AVR	36
Gambar 20. Pengaturan <i>Communication</i> Pada Bascom AVR	36
Gambar 21. Hasil Pengujian Sistem Minimum dan LCD Grafik	45

DAFTAR LAMPIRAN

Lampiran 1. Lembar Masukan.....	56
Lampiran 2. Gambar Rangkaian.....	58
Lampiran 3. Data sheet LCD grafik (Biru)	61
Lampiran 4. Data Sheet ATmega32	76
Lampiran 5. Program Utama	85

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Anak dengan berkebutuhan khusus membutuhkan suatu pola layanan tersendiri, khususnya bagi anak tuna rungu yang memiliki keterbatasan dalam berkomunikasi. Secara umum anak tuna rungu mempunyai tingkah laku, kebiasaan dan kepribadian yang sama dengan anak normal lainnya. Cara untuk berfikir dan belajar mereka juga sama hanya terhambat oleh pendengaran saja, maka dibutuhkan media untuk mereka berkomunikasi.

Selama ini cara berkomunikasi mereka kebanyakan melalui indera visual karena mereka memiliki daya tangkap visual yang tinggi, salah satunya menggunakan bahasa isyarat jari. Beberapa cara digunakan anak tuna rungu untuk berkomunikasi seperti menggunakan gambar atau menulis. Namun cara berkomunikasi dengan menulis atau menggambar membutuhkan waktu beberapa saat dan dirasa cukup lama, maka cara yang lebih efektif adalah menggunakan bahasa isyarat atau lebih jelasnya menggunakan isyarat jari.

Untuk dapat berkomunikasi dengan bahasa isyarat jari, mereka perlu menghafal abjad jari untuk membentuk suatu kata dan kalimat. Banyak media visual yang dapat digunakan untuk belajar anak tuna rungu mempelajari abjad jari. Namun media visual yang digunakan terkesan monoton dan tidak didesain khusus sehingga membuat anak tuna rungu kurang tertarik dalam mempelajarinya.

Maka dari itu dirancanglah sebuah alat dengan teknologi yang sedang berkembang saat ini, sehingga dapat diaplikasikan dalam kehidupan sehari-hari. Prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 merupakan alat yang diaplikasikan sebagai media untuk mempelajari huruf abjad jari.

Prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 menggunakan keypad dan LCD grafik. Keypad A sampai Z digunakan sebagai input huruf abjad dan LCD grafik digunakan sebagai display karakter abjad jari.

Alat – alat pengenalan abjad jari yang selama ini digunakan merupakan media klasik. Maka perlu dikembangkan sebuah media pengenalan abjad jari yang memiliki kelebihan dibandingkan dengan media klasik. Prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 memiliki kelebihan diantaranya dapat digunakan secara praktis dan portable.

Pengendali prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 adalah mikrokontroller AVR ATmega 32. Mikrokontroller keluarga AVR ini merupakan keluaran dari produsen mikrokontroller Atmel. Pengendali mikrokontroller ATmega 32 memiliki kelebihan yaitu lebih flexible dan bisa mengurangi penggunaan komponen - komponen elektronika yang biasa digunakan pada produk dari keluarga MCS-51. Kelebihan lain sebagai pengendali utama karena mikrokontroller ini mempunyai kecepatan eksekusi yang tinggi dibandingkan mikrokontroller keluarga MCS-51.

B. Identifikasi Masalah

Dari uraian latar belakang, maka dapat dibuat suatu identifikasi masalah sebagai berikut :

1. Anak tuna rungu memiliki keterbatasan dalam berkomunikasi.
2. Banyak media visual pengenalan abjad jari namun tidak didesain khusus untuk tuna rungu.
3. Alat – alat pengenalan abjad jari yang selama ini digunakan merupakan media klasik.
4. Belum ada media untuk tuna rungu tentang pengenalan abjad jari dengan keypad sebagai input dan LCD grafik sebagai penampil karakter abjad jari.
5. Belum dimanfaatkannya mikrokontrol ATmega 32 sebagai prototipe pengenalan abjad jari untuk tuna rungu yang modern, praktis dan mudah digunakan.
6. Kurang optimalnya penggunaan aplikasi elektronika pada media belajar tuna rungu.

C. Batasan Masalah

Berdasarkan latar belakang dan identifikasi masalah, perlu adanya batasan masalah sehingga ruang lingkup masalah menjadi lebih jelas. Adapun batasan masalah yang diambil yaitu pembuatan prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32, yang menggunakan keypad A sampai Z sebagai input huruf abjad dan LCD grafik sebagai media penampil karakter huruf abjad jari.

D. Rumusan Masalah

Dari identifikasi yang ada, maka dapat ditarik beberapa rumusan masalah, yaitu :

1. Bagaimana merancang *hardware* prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32?
2. Bagaimana mengembangkan program prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32?
3. Bagaimana unjuk kerja prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32?
4. Bagaimana tanggapan dari pakar tuna rungu tentang prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32?

E. Tujuan

Adapun tujuan dari pembuatan prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 yaitu:

1. Merealisasikan rancangan *hardware* prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32.
2. Merealisasikan penggunaan *software* prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32.
3. Mengetahui unjuk kerja prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32.
4. Mengetahui tanggapan dari pakar tuna rungu tentang prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32

F. Manfaat

Pembuatan proyek akhir ini diharapkan dapat bermanfaat bagi mahasiswa, lembaga pendidikan, dan industri. Adapun manfaat yang diharapkan dari pembuatan tugas akhir ini antara lain :

1. Bagi mahasiswa
 - a. Untuk mengaplikasikan ilmu yang didapat selama di bangku kuliah dan menerapkan ilmunya secara nyata.
 - b. Dapat digunakan sebagai bahan referensi atau pembelajaran dan penambah wawasan tentang aplikasi LCD grafik yang berbasis mikrokontroler khususnya ATmega 32 serta sebagai kajian untuk pengembangan selanjutnya.
2. Bagi jurusan
 - a. Sebagai wujud dari perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK).
 - b. Sebagai parameter kualitas dan kuantitas lulusan mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta.
3. Bagi Dunia Pendidikan

Dapat digunakan sebagai alat bantu untuk tuna rungu dalam mempelajari pengenalan abjad jari.

G. Keaslian Gagasan

Prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 ini dibuat karena sepengetahuan penulis belum ada yang membuat

di Universitas Negeri Yogyakarta. Dan sepengetahuan penulis alat yang sudah ada tidak menggunakan ATmega 32 melainkan menggunakan AT89S53.

Berlanjut dari hal tersebut penulis membuat prototipe pengenalan abjad jari untuk tuna rungu berbasis ATmega 32 dengan konsep yang berbeda dari yang sudah ada.

Adapun karya sejenis yang berkaitan dengan proyek akhir ini adalah Tugas Akhir Pembuatan display digital untuk modul kompas CMPS – 03 oleh Panca Wahyudi Dari Universitas Kristen Petra Surabaya. Perbedaan dengan alat yang penulis buat adalah mikrokontroler yang digunakan, yaitu memakai ATmega 32 sebagai pusat kendali seluruh system dan LCD grafik digunakan sebagai penampil karakter abjad jari.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Tuna Rungu

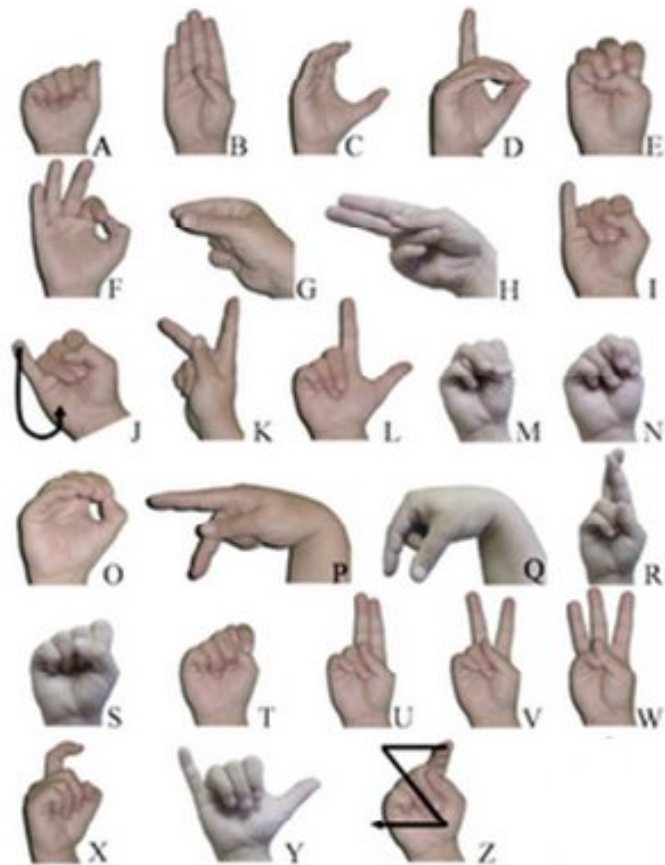
Tuna rungu dapat diartikan sebagai suatu keadaan kehilangan pendengaran. Anak tunarungu adalah anak yang mengalami kekurangan dari segi pendengaran sehingga memerlukan pelayanan khusus. Menurut Dwidjosumarto (Soemantri, 1995:140) bahwa “seseorang yang tidak atau kurang mampu mendengar suara dikatakan tuna rungu”. Sedangkan Soemantri (1995:145) mengatakan bahwa “tuna rungu adalah sebagai suatu keadaan kehilangan pendengaran yang mengakibatkan seseorang tidak dapat menangkap berbagai rangsangan, terutama melalui indera pendengarannya”.

B. Metode Abjad Jari

Secara harfiah, abjad jari merupakan usaha untuk menggambarkan alphabet secara manual dengan menggunakan satu tangan. Abjad jari adalah isyarat yang dibentuk dengan jari-jari tangan (tangan kanan atau tangan kiri) untuk mengeja huruf atau angka. Abjad jari digunakan untuk:

- a. Mengisyaratkan nama diri
- b. Mengisyaratkan singkatan atau akronim; dan
- c. Mengisyaratkan kata yang belum ada isyaratnya

Prototipe ini ditujukan untuk anak tuna rungu dalam tahap awal belajar tentang pengenalan huruf abjad. Dalam pelaksanaanya, awalnya anak membutuhkan pendamping untuk menggunakan prototipe ini. Selebihnya anak tersebut dapat belajar dan menggunakannya sendiri.

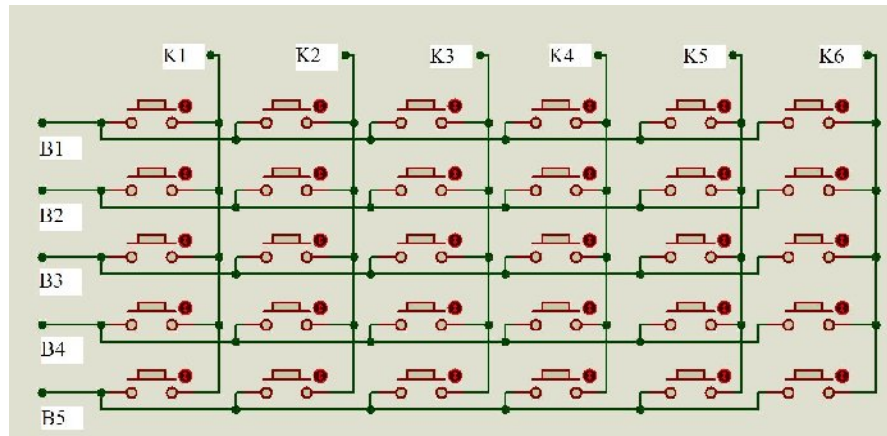


Gambar 1. Abjad jari

C. Perangkat Keras

1. Keypad Matriks

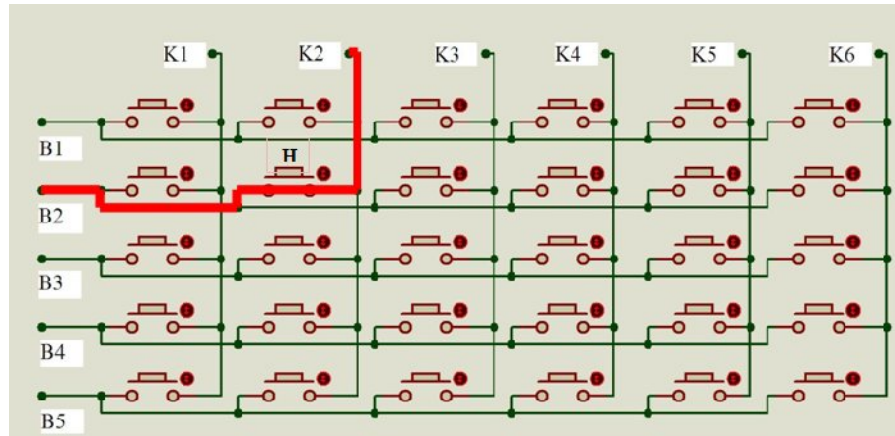
Keypad Matriks adalah tombol-tombol yang disusun secara matriks (baris x kolom) sehingga dapat mengurangi penggunaan pin input. Sebagai contoh, Keypad Matriks 5×6 cukup menggunakan 11 pin untuk 30 tombol. Hal tersebut dimungkinkan karena rangkaian tombol disusun secara horizontal membentuk baris dan secara vertikal membentuk kolom.



Gambar 2. Skema Keypad Matriks

Namun demikian, sebagai konsekuensi dari penggunaan bersama satu jalur (semisal baris satu (B1)), maka tidak dimungkinkan pengecekan dua tombol sekaligus dalam satu slot waktu.

Proses pengecekan dari tombol yang dirangkai secara maktriiks adalah dengan teknik *scanning*, yaitu proses pengecekan yang dilakukan dengan cara memberikan umpan-data pada satu bagian dan mengecek *feedback* (umpan-balik) – nya pada bagian yang lain. Dalam hal ini, pemberian umpan-data dilakukan pada bagian baris dan pengecekan umpan-balik pada bagian kolom. Pada saat pemberian umpan-data pada satu baris, maka baris yang lain harus dalam kondisi inversi-nya. Tombol yang ditekan dapat diketahui dengan melihat asal data dan di kolom mana data tersebut terdeteksi:



Gambar 3. Cara kerja Keypad Matriks

Pada contoh di atas, tombol yang ditekan adalah tombol “H”. Seperti terlihat bahwa B2 bernilai nol, sedangkan B1, B3, B4 dan B5 adalah satu. Kemudian dengan mengetahui bahwa asal data dari B2, dan umpan-baliknya terdeteksi pada K2, maka dapat disimpulkan bahwa tombol yang ditekan adalah tombol “H”.

2. *Graphic Liquid Crystal Display 128 X 64*

Graphic liquid crystal display ini memiliki titik sebanyak 128 X 64, sehingga dapat digunakan untuk menampilkan karakter gambar yang bervariasi. Dimensi dari Graphic LCD tipe WG12864C-TMI dan besarnya titik atau *pixel* pada LCD.

Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)

- Pin 1 (CS1) merupakan *chip select* untuk kolom 1 sampai 64
- Pin 2 (CS2) merupakan *chip select* untuk kolom 64 sampai 128
- Pin 3 (GND) dan pin 4 (VDD) terhubung pada *power supply* dengan tegangan input 5 volt.
- Pin 5 (Vo) berfungsi sebagai pengatur *contrast* pada *display* LCD. Untuk pengaturannya diperlukan sebuah potensiometer yang terhubung dengan *ground* dan pin 18 VEE.
- Pin 6 (D/I) berfungsi untuk memilih perintah melakukan data atau instruksi. Jika memilih *high*, maka pin akan berfungsi sebagai data. Apabila memilih *low*, maka pin akan berfungsi sebagai instruksi.
- Pin 7 (R/W) berfungsi untuk memberikan kondisi pada *graphic* LCD. Jika *high* maka *graphic* LCD melakukan pembacaan data. Jika *low* maka *graphic* LCD sebagai *write*.

- g. Pin 8 (E) *enable signal* berfungsi untuk mengaktifkan fungsi program.
- h. Pin 9 – 16 (DB0 – DB7) merupakan 8 bit untuk input data.
- i. Pin 17 (RST) berfungsi sebagai *reset* program.
- j. Pin 18 (VEE) menghasilkan tegangan -5 volt, berfungsi memberikan *power supply* pin 5 sebagai pengatur *contrast* LCD.
- k. Pin 19 dan 20 dihubungkan dengan *power supply*. Untuk pin 19 dihubungkan pada positif *power supply* dan pin 20 dihubungkan ke *ground*.

Tabel 1. Fungsi masing – masing pin *graphic* LCD

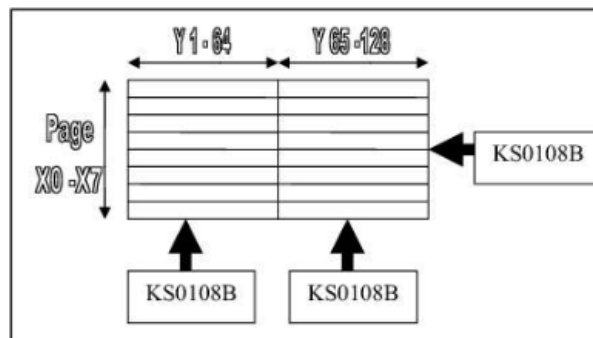
Pin	Symbol	Level	Description
1	$\overline{\text{CS1}}$	L	Select Segment 1 ~ Segment 64
2	$\overline{\text{CS2}}$	L	Select Segment 65 ~ Segment128
3	GND	0V	Ground
4	VDD	5.0V	Supply voltage for logic
5	V_{LC}/VO	(Variable)	Operating voltage for LCD
6	D/I	H/L	H: Data , L: Instruction
7	R/W	H/L	H: Read(MPU←Module) , L :Write(MPU→Module)
8	E	H	Enable signal
9	DB0	H/L	Data bit 0
10	DB1	H/L	Data bit 1
11	DB2	H/L	Data bit 2
12	DB3	H/L	Data bit 3
13	DB4	H/L	Data bit 4
14	DB5	H/L	Data bit 5
15	DB6	H/L	Data bit 6
16	DB7	H/L	Data bit 7

Lanjutan tabel 1. Fungsi masing – masing pin *graphic* LCD

Pin	Symbol	Level	Description
17	RST	L	Reset the LCM
18	VEE	V	Negative voltage;
19	A	—	Power supply for LED +
20	K	—	Power supply for LED -

Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)

Pada *graphic* LCD ini menggunakan 3 buah IC *driver* KS0108B (64CH Segment Driver For Dormatrix LCD) yang memiliki fungsi untuk membentuk dot sebanyak 128x64. Untuk IC pertama berfungsi mengatur banyaknya kolom 1 sampai 64, sedangkan IC kedua mengatur kolom 65 sampai 128. Untuk IC ketiga berfungsi untuk mengatur baris 1 sampai 64. Untuk pembagian kerja IC KS0108B dan keseluruhan blok diagram dari *graphic* LCD adalah sebagai berikut:



Gambar 5. Pembagian IC KS0108B

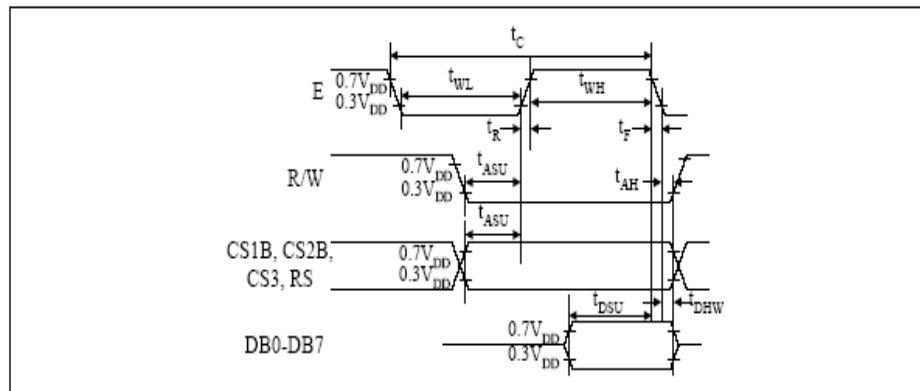
Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)

Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)

yaitu bagian kiri (kolom 1 sampai 64) dan bagian kanan (kolom 65 sampai 128).

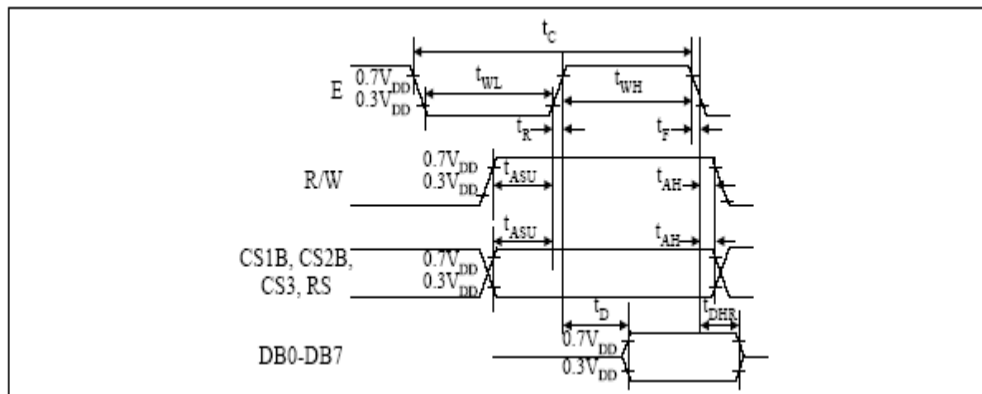
- Mengaktifkan display
- Mengaktifkan bagian display (kolom 1 - 64 atau kolom 65 – 128)
- Mengatur letak Y
- Mengatur letak X
- Mengatur mulai dari sumbu Y
- Mengirim Instruksi
- Mengirim data
- Menampilkan

Grafik LCD memiliki dua kondisi yaitu *read* dan *write*. Untuk melakukan tampilan ke LCD menggunakan instruksi *write*, sedangkan untuk pembacaan dari LCD menggunakan instruksi *read*.



Gambar 7. Urutan *Write Timing*

Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)



Gambar 8. Urutan *Read Timing*

Sumber: (<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc>)

Untuk pengoperasian LCD ini langkah awal adalah mengirim instruksi lalu kemudian mengirim data. Untuk membedakan antara instruksi dan data adalah pin D/I. Jika diberi logic 0, maka akan berfungsi sebagai

instruksi. Jika logic 1, maka berfungsi sebagai data, CS1 dan CS2 berfungsi untuk mengaktifkan sisi kanan atau sisi kiri yang aktif.

3. Mikrokontroler AVR

a. Arsitektur ATmega 32

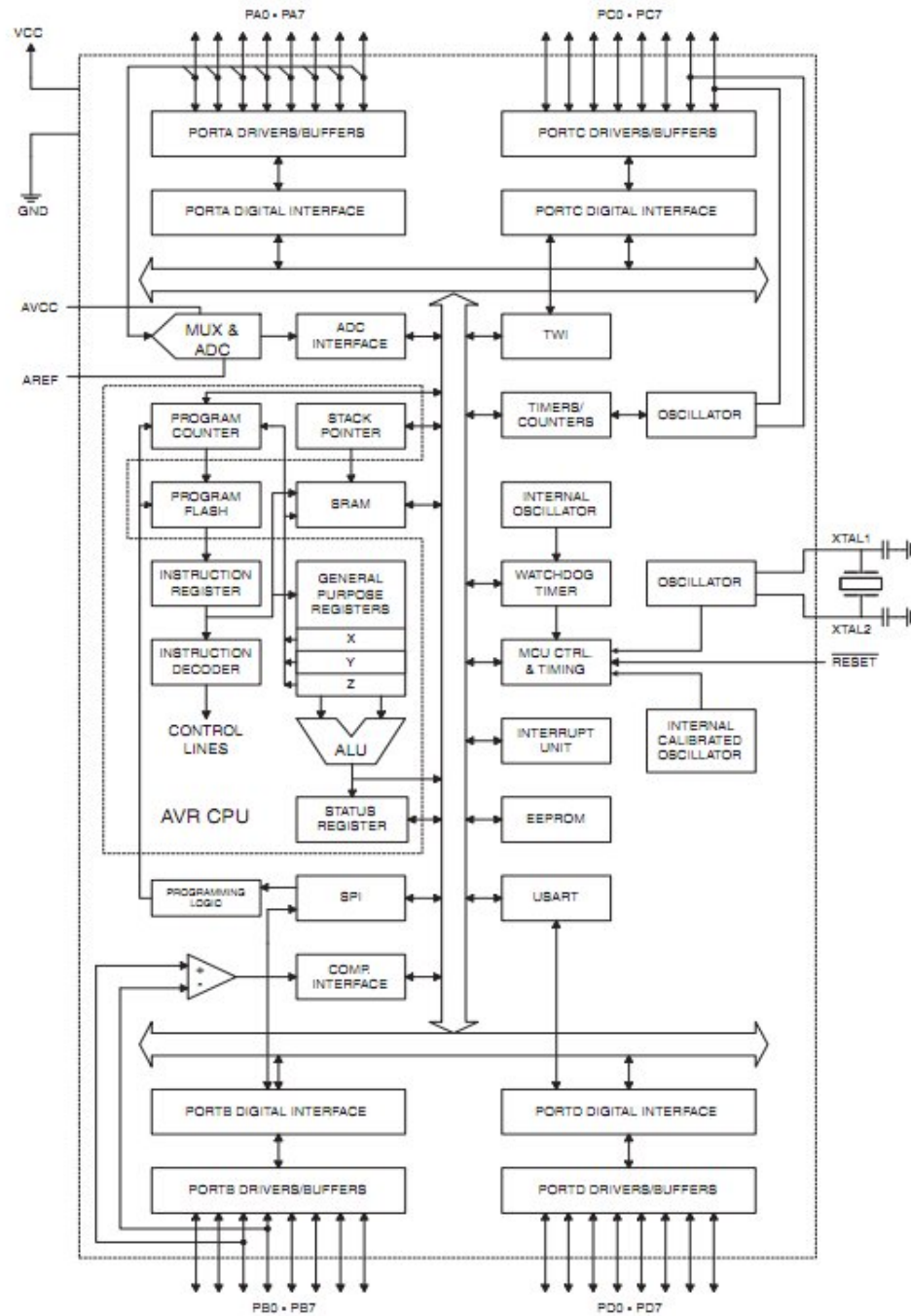
Mikrokontroller dapat dianalogikan seperti sebuah sistem komputer yang dikemas dalam sebuah chip. Dalam sebuah chip mikrokontroller sudah terdapat kebutuhan minimal agar mikroprosesor dapat bekerja, yaitu meliputi mikroprosesor, ROM, RAM, I/O, dan *clock* seperti yang dimiliki sebuah Personal Komputer (PC).

Karena ukurannya yang relatif kecil membuat mikrokontroller menjadi lebih fleksibel dan praktis digunakan terutama pada sistem yang tidak terlalu kompleks dan tidak membutuhkan beban komputasi yang tinggi.

b. Fitur ATmega 32

Fitur – fitur yang dimiliki ATmega 32 sebagai berikut :

- 1) Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
- 2) Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz.
- 3) Memiliki kapasitas *Flash* memori 32 KByte, EEPROM 1 KByte dan SRAM 2 KByte.

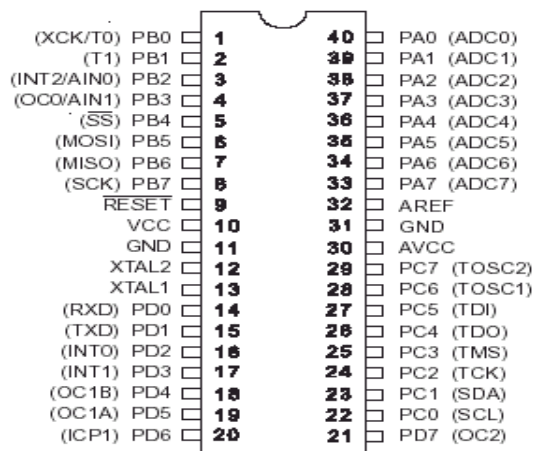


Gambar 9. Blok Diagram AVR ATmega 32

(<http://www.atmel.com>)

- 4) Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
- 5) CPU yang terdiri atas 32 buah *register*.
- 6) Unit interupsi *internal* dan *eksternal*.
- 7) *Port* USART untuk komunikasi serial.
- 8) Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, mode *Compare* dan mode *Capture*.
- 9) Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan mode *Compare*.
- 10) *Real Time Counter* dengan *Oscillator* tersendiri.
- 11) 4 *Channel* PWM
- 12) 8 *Channel* 10 bit ADC

c. Konfigurasi Pin Atmega 32



Gambar 10. Konfigurasi Pin ATmega 32

(<http://www.atmel.com>)

Konfigurasi pin ATmega 32 dengan kemasan 40 pin DIP (*Dual In-line Package*). Dari gambar 10 dapat dijelaskan fungsi dari masing-masing pin ATmega 32 sebagai berikut :

1) Port A (PA0 – PA7)

Port A adalah 8-bit port I/O yang bersifat bi-directional dan setiap pin memiliki internal pull-up resistor. Output buffer port A dapat mengalirkan arus sebesar 20 mA. Ketika port A digunakan sebagai input dan di pull-up secara langsung, maka port A akan mengeluarkan arus jika internal pull-up resistor diaktifkan. Pin-pin dari port A memiliki fungsi khusus yaitu dapat berfungsi sebagai channel ADC (Analog to Digital Converter) sebesar 10 bit.

2) Port B (PB0 - PB7)

Port B adalah 8-bit port I/O yang bersifat bi-directional dan setiap pin mengandung internal pull-up resistor. Output buffer port B dapat mengalirkan arus sebesar 20 mA. Ketika port B digunakan sebagai input dan di pull-down secara external, port B akan mengalirkan arus jika internal pull-up resistor diaktifkan.

Pin-pin port B memiliki fungsi-fungsi khusus, diantaranya :

- SCK port B, bit 7

Input pin clock untuk up/downloading memory.

- MISO port B, bit 6

Pin output data untuk uploading memory.

- MOSI port B, bit 5

Pin input data untuk downloading memory.

3) Port C (PC0 – PC7)

Port C adalah 8-bit port I/O yang berfungsi bi-directional dan setiap pin memiliki internal pull-up resistor. Output buffer port C dapat mengalirkan arus sebesar 20 mA. Ketika port C digunakan sebagai input dan di pull-down secara langsung, maka port C akan mengeluarkan arus jika internal pull-up resistor diaktifkan.

4) Port D (PD0 – PD7)

Port D adalah 8-bit port I/O yang berfungsi bi-directional dan setiap pin memiliki internal pull-up resistor. Output buffer port D dapat mengalirkan arus sebesar 20 mA. Ketika port D digunakan sebagai input dan di pull-down secara langsung, maka port D akan mengeluarkan arus jika internal pull-up resistor diaktifkan.

5) VCC merupakan pin yang berfungsi sebagai masukan catu daya.

6) GND merupakan pin Ground.

7) **RESET**: merupakan input reset yang bekerja pada level rendah (*active low*) selama minimal 1,5us.

8) **XTAL1** : *Input* ke penguat *inverting* oscillator dan input ke *internal. clock*

9) **XTAL2** : *Output* dari penguat *inverting* oscillator.

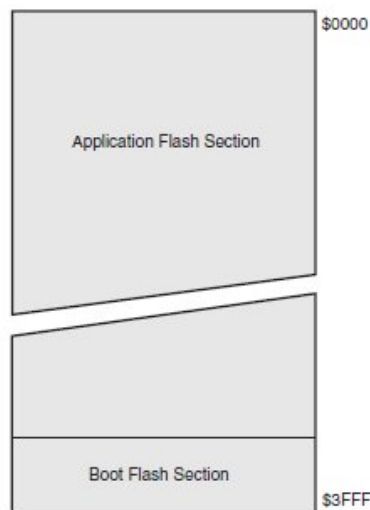
10) **AVCC** merupakan catu daya yang digunakan sebagai masukan analog ADC yang terhubung ke port A.

11) **AREF** merupakan tegangan referensi analog untuk ADC.

d. Peta Memori

1) *Memori Flash*

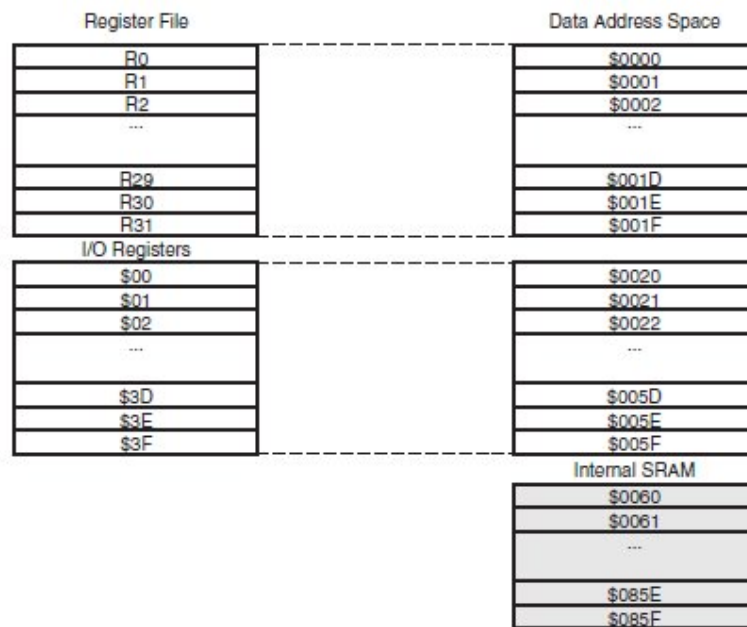
Arsitektur AVR mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu ATmega 32 memiliki memori EEPROM untuk menyimpan data. ATmega 32 memiliki 32 Kbyte On-chip In-system Reprogrammable Flash Memory untuk menyimpan program. Karena semua instruksi AVR memiliki format 16 atau 32 bit, Flash diatur dalam 8K x 16 bit. Untuk keamanan program, memori program, flash dibagi kedalam dua bagian yaitu bagian program boot dan aplikasi. Bootloader adalah program kecil yang bekerja pada saat start up time yang dapat memasukan seluruh program aplikasi kedalam memori prosesor.



Gambar 11. Peta Program memori
<http://www.atmel.com>

2) Memori Data

Gambar 11 menunjukkan peta memori SRAM pada Atmega 32. Terdapat 2144 lokasi *address* data memori. 96 lokasi *address* digunakan untuk *Register File* dan *I/O Memory* ,selanjutnya 2024 lokasi *address* lainnya digunakan untuk internal data SRAM. *Register File* terdiri dari 32 *General Purpose Register* (GPR), *I/O register* terdiri dari 64 *register*.



Gambar 12. Peta Data Memori

(<http://www.atmel.com>)

Memori data AVR ATmega 32 terbagi menjadi 3 bagian yaitu 32 buah register umum, 64 buah register I/O dan 2 Kbyte SRAM internal. General purpose register menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64

alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral.

3) Memory EEPROM

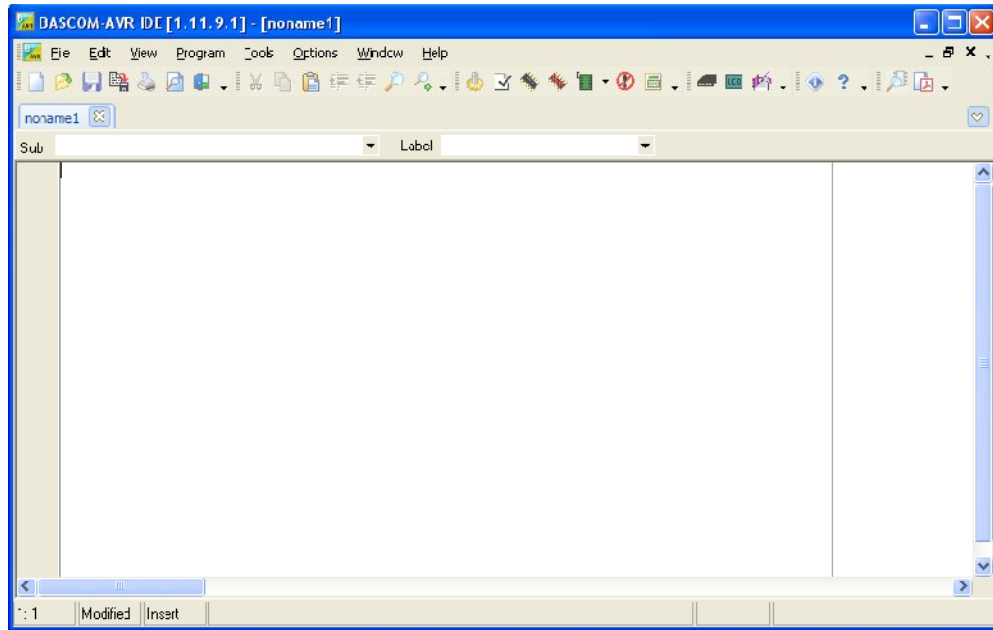
ATmega 32 memiliki memori EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1024 byte yang terpisah dari memori program maupun memori data. Memori EEPROM bisa digunakan untuk menyimpan data yang dapat bertahan atau tersimpan walaupun mikrokontroler tanpa tegangan catu daya atau tahan terhadap gangguan catu daya. Memori EEPROM ini hanya bisa diakses dengan menggunakan *register I/O*

e. Perangkat Lunak (*Software*)

1) Bascom Avr

BASCOM-AVR adalah program *basic compiler* berbasis windows untuk mikrokontroler keluarga AVR merupakan pemrograman dengan bahasa tingkat tinggi ” *BASIC* ” yang dikembangkan dan dikeluarkan oleh MCS elektronika sehingga dapat dengan mudah dimengerti atau diterjemahkan. Dalam program BASCOM-AVR terdapat beberapa kemudahan, untuk membuat program *software* ATmega 32, seperti program simulasi yang sangat berguna untuk melihat, simulasi hasil program yang telah kita buat, sebelum program tersebut kita *download* ke IC atau ke mikrokontroler.

Ketika program BASCOM-AVR dijalankan dengan mengklik icon BASCOM-AVR, maka jendela berikut akan tampil :



Gambar 13. Tampilan Jendela Program BASCOM AVR

Sumber : (<https://fahmizaleeits.wordpress.com>)

BASCOM-AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika kita membuat aplikasi yang berhubungan dengan LCD

2) Tipe data

Tipe data merupakan bagian program yang paling penting karena akan mempengaruhi setiap instruksi yang digunakan. Dapat dilihat pada tabel .2

Tabel 2. Tipe-tipe data

NO	Tipe	Nomor Pin	Jangkauan
1	Bit	1	0 atau 1
2	Byte	2	0-255
3	Integer	3	-32,768-32,767
4	Word	4	0-65535
5	Long	5	-2147483648-2147483647
6	Single	6	$1.5 \times 10^{(-45)}$ - 3.4×10^{38}
7	Double	7	5.0×10^{324} – 1.7×10

3) Deklarasi

Deklarasi diperlukan bila programmer akan menggunakan pengenalan (*indentifier*) dalam program. *Identifier* dapat berupa variabel, konstanta dan fungsi.

4) Operator

Terdapat lima operator dalam Bahasa Basic yaitu operator penugasan, operator aritmatika, operator perbandingan, operator logika dan operator bitwise. Setiap operator memiliki fungsi masing-masing sesuai dengan nama operator yang akan digunakan.

5) Komentar Program

Komentar program diperlukan untuk memudahkan pembacaan dan pemahaman suatu program.

6) Penyeleksaian Kondisi

Penyeleksian kondisi digunakan untuk membandingkan dan mengarahkan alur suatu proses program. Struktur kondisi yang dapat digunakan diantaranya “If..”, “If..Else”, dan “Case”

7) Perulangan

Dalam Bahasa Basic tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak nilai yang telah ditentukan sebelumnya. Struktur pengulangan tersebut mempunyai bentuk yang bermacam-macam seperti “While”, “Do..Loop” dan “For”.

Berikut ini beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler ATmega 32.

Tabel 3. Instruksi dasar Bascom AVR

Instruksi	Keterangan
<i>DO....LOOP</i>	Perulangan
<i>GOSUB</i>	Memanggil Prosedur
<i>IF....THEN</i>	Percabangan
<i>FOR....NEXT</i>	Perulangan
<i>WAIT</i>	Waktu Tunda Detik
<i>WAITMS</i>	Waktu Tunda Mili Detik
<i>WAITUS</i>	Waktu Tunda Micro Detik
<i>GOTO</i>	Loncat Kealamat Memori
<i>SELECT....CASE</i>	Percabangan

BAB III

KONSEP RANCANGAN

A. Identifikasi kebutuhan

Dalam merancang sebuah Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu Berbasis Mikrokontroller ATmega 32 dibutuhkan beberapa komponen yang terdiri dari :

1. Adaptor *step down* untuk mencatu daya ke alat.
2. Keypad Matriks untuk memberikan input data.
3. Sistem minimum Atmega 32 untuk memproses data dari input ke output.
4. LCD Grafik sebagai penampil karakter abjad jari.

B. Analisis Kebutuhan

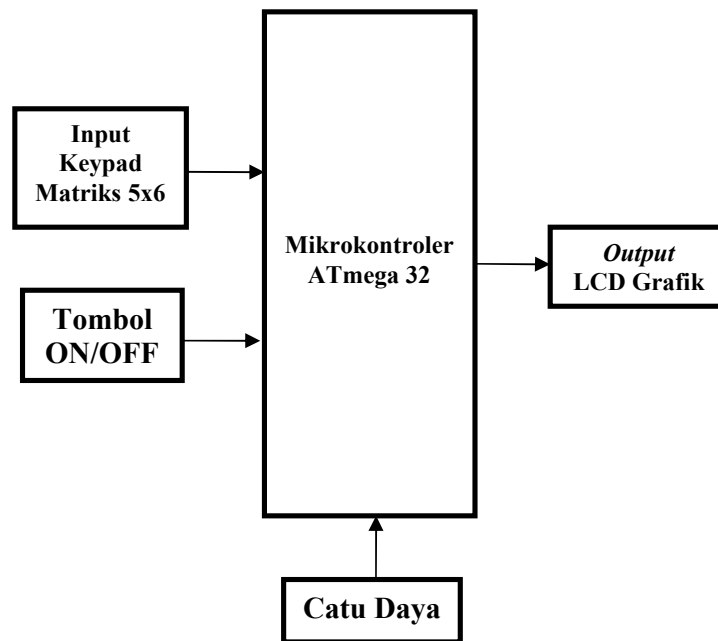
Berdasarkan identifikasi kebutuhan yang ada, maka diperlukan beberapa spesifikasi dari komponen atau rangkaian sebagai berikut :

1. Catu daya yang menghasilkan tegangan rangkaian sebesar 4,7 - 12 volt langsung dari trafo dan sudah disearahkan menjadi 5 Volt DC untuk sistem minimum ATmega 32 dari rangkaian regulator 7805.
2. Keypad Matriks 5x6 untuk memberikan input dalam bentuk huruf.
3. Menggunakan mikrokontroler ATmega 32 sebagai unit pengendali utama untuk menangani semua proses sistem.
4. LCD Grafik sebagai output berupa karakter gambar abjad jari.
5. Saklar ON/OFF digunakan untuk menghidupkan dan mematikan sistem.
6. Box (*casing*) digunakan untuk pemasangan rangkaian dari prototipe.

C. Perancangan Alat

1. Perancangan Perangkat Keras (*Hardware*)

Rangkaian elektronik alat ini terdiri atas beberapa blok, diantaranya sebagai berikut :



Gambar 14. Diagram blok sistem prototype alat

a. Cara kerja blok input

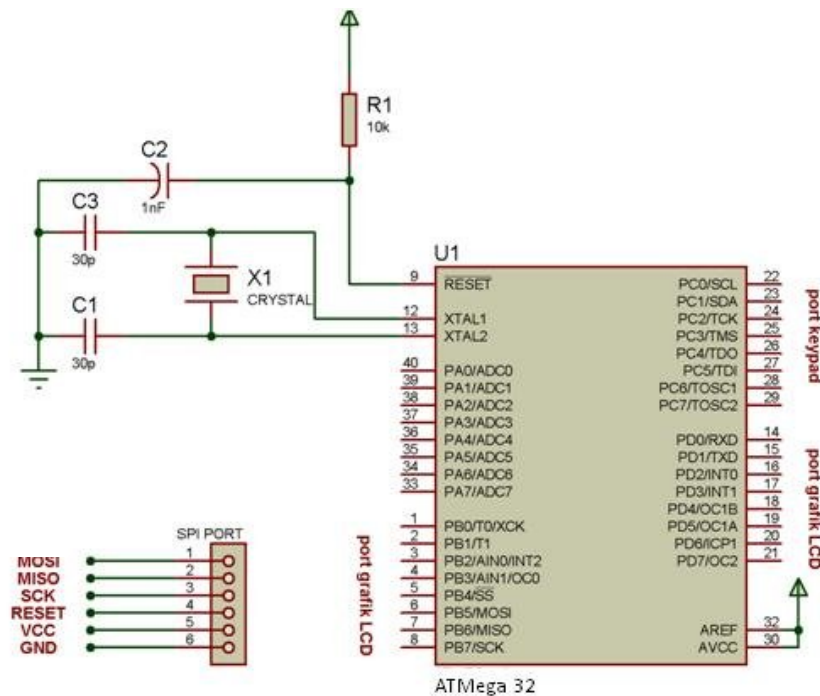
Blok input disini menggunakan keypad matrik 5x6 dengan 27 buah push button yang disusun secara matrik sehingga memiliki indeks baris dan kolom. Keypad ini didesain sendiri sesuai dengan kebutuhan input abjad jari.

Selain inputan keypad, pada blok ini juga terdapat tombol *On/Off*. Tombol *On/Off* digunakan untuk menghidupkan dan mematikan sistem.

b. Mikrokontroller ATmega 32

Sistem kontrol yang digunakan adalah sistem minimum ATmega 32. Sistem minimum adalah kebutuhan minimal yang harus dipenuhi agar mikrokontroller dapat bekerja dengan normal. Sistem minimum terdiri dari bagian *reset*, *clock*, *port input output* dan *port ISP* (*In system programming*) .

Dengan adanya port ISP maka tidak dibutuhkan lagi rangkaian *downloader* khusus untuk memasukkan program ke dalam Mikrokontroller. Program bisa langsung di *upload* ke dalam mikrokontroller menggunakan parallel port komputer atau PC.



Gambar 15. Rangkaian sistem minimum ATmega 32

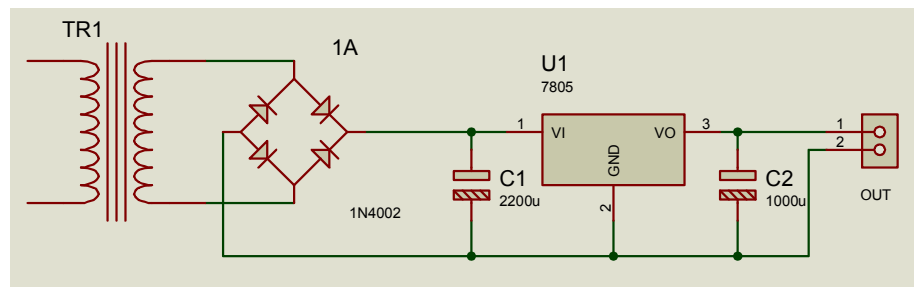
c. Blok output

Pada sistem ini digunakan LCD grafik karena LCD jenis ini memiliki titik sebanyak 128x64, sehingga dapat digunakan untuk menampilkan karakter huruf dan gambar yang bervariasi.

2. Perencanaan Rangkaian

a. Catu Daya

Rangkaian catu daya sangat penting karena tanpa catu daya alat ini tidak dapat bekerja. Rangkaian catu daya digunakan untuk menyuplai tenaga ke beberapa rangkaian seperti sistem minimum ATmega 32, keypad dan LCD grafik.



Gambar 16. Rangkaian catu daya

Sistem minimum memerlukan tegangan maksimal 5 volt untuk bisa bekerja optimal tanpa merusak mikrokontroler.

Gambar 17. Rangkaian prototipe pengenalan abjad jari

3. Langkah Pembuatan alat

- a. Alat yang akan digunakan dalam pembuatan alat ini adalah :

Tabel 4. Daftar alat yang dibutuhkan

No	Nama Bahan	Jumlah
1.	Multimeter	1 buah
2.	Tool set	1 set
3.	Tang cucut	1 buah
4.	Bor listrik	1 buah
5.	Solder	1 buah
6.	Tang potong	1 buah
7.	Obeng	1 buah
8.	Gunting	1 buah
9.	Pinset	1 buah
10.	Atraktor	1 buah

- b. Bahan yang akan digunakan dalam pembuatan alat ini adalah :

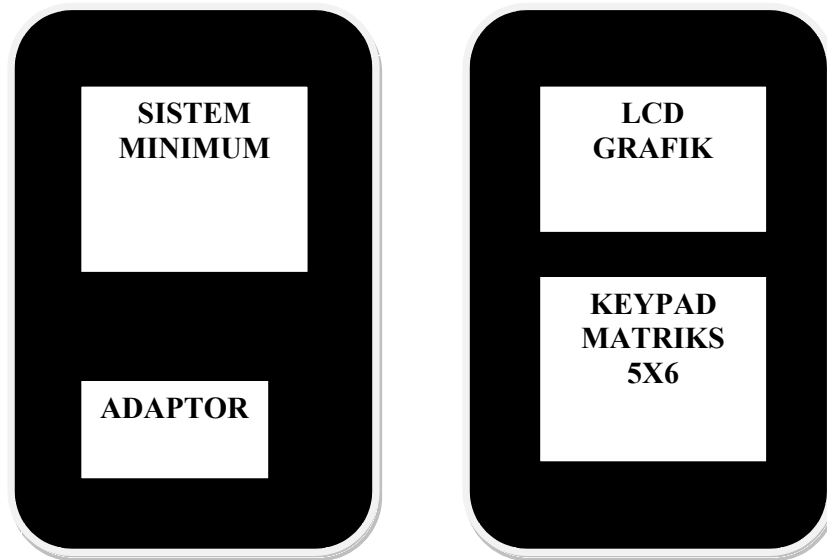
Tabel 5. Daftar komponen yang dibutuhkan

No	Nama Bahan	Jumlah
1.	Chasing	1 buah
2.	Sekrup	Secukupnya
3.	Push button	Secukupnya
4.	Kabel penghubung	Secukupnya
5.	LCD Grafik	1 Buah
6.	Keypad matrik	1 Buah
7.	Kapasitor	Secukupnya
8.	Resistor	Secukupnya
9.	PCB	Secukupnya
10.	Ferry Chloride	Secukupnya
11.	Tenol	Secukupnya
12.	Dioda	1 Buah

- c. Spesifikasi alat

Prototipe pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 ini mempunyai spesifikasi alat sebagai berikut:

- 1) Tegangan *input* maksimal 7,5 volt DC dan minimum 3 volt.
- 2) LCD grafik 128 x 64
- 3) *Keypad matrix* 5 x 6
- 4) Box (*casing*) dengan panjang 11,5 cm dan lebar 9,5 cm



Bagian dalam box

Bagian depan box

Gambar 18. Bagian Box

d. Pembuatan *Layout*

Dalam penyelesaiannya menjadi sebuah alat, harus melalui beberapa langkah kerja sebagai berikut:

- 1) Membuat gambar rangkaian pada program ISIS Profesional
- 2) Merancang *layout* pada komputer menggunakan ARES 7 Profesional
- 3) Mengkopi *layout* ke media kertas *Gloosy*
- 4) Mensablon *layout* ke PCB

5) Melarutkan tembaga PCB dengan menggunakan larutan

FerryClorida

6) Mengumpulkan komponen yang kondisinya baik dan bahan-bahan yang diperlukan

7) Pengeboran PCB, pemasangan, penyolderan.

8) Membuat program dengan bahasa *basic*

9) Uji coba tiap blok rangkaian untuk mengetahui karakteristik fungsinya

10) Uji coba keseluruhan alat untuk kinerja alat sehingga sesuai dengan yang diharapkan.

11) Mengamati cara kerja rangkaian

12) Menganalisa hasil pengujian

13) Selesai.

e. Pembuatan PCB

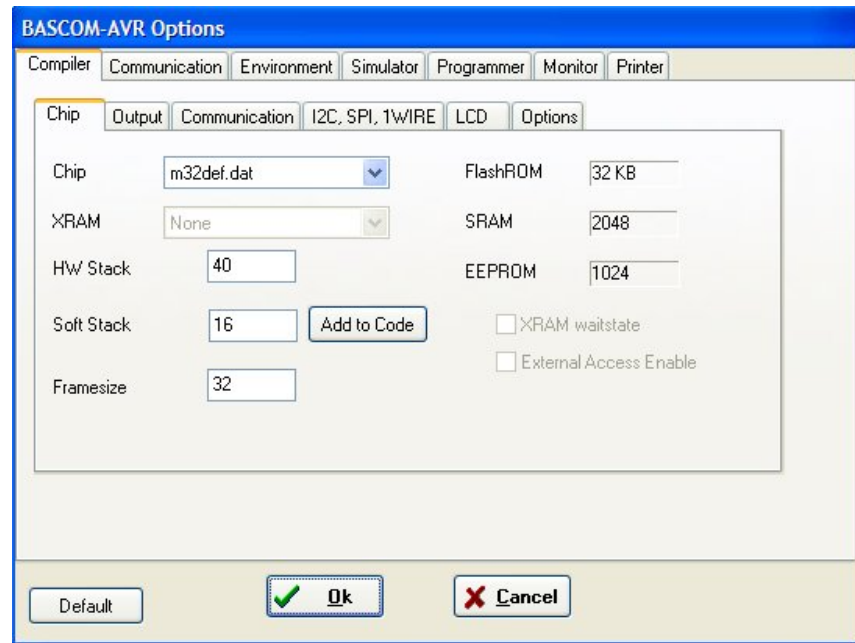
Pembuatan PCB bisa dilakukan dengan banyak cara yaitu menggambar langsung ke papan PCB menggunakan spidol permanen, menggunakan teknik sablon, di glossy dan sebagainya. Langkah pertama dalam pembuatan PCB yaitu dengan membuat *Layout* PCB menggunakan ARES (*Advanced Routing and Editing Software*). ARES merupakan software khusus yang digunakan dalam mendesain PCB (*Printed Circuit Board*) yang termasuk dalam salah satu modul dari Software Proteus. Langkah – langkah dalam pembuatan PCB menggunakan ARES adalah :

- 1) Membuat papan PCB menggunakan *Board Edge*.
- 2) Memilih *package* komponen sesuai jenis komponen dengan memilih *package* mode pada *toolbar*.
- 3) Memasukkan semua *package* komponen ke dalam PCB Board Edge.
- 4) Mengatur tata letak komponen di dalam PCB.
- 5) Menghubungkan kaki – kaki komponen sesuai rangkaian menggunakan *ratsnest* mode pada *toolbar*.
- 6) Membuat jalur menggunakan *auto router* yang terdapat pada menu *tools*.
- 7) *Print layout* PCB dengan spesifikasi *layers = bottom copper* dan *top copper* , *scale* 100% , dan *reflection* = normal.

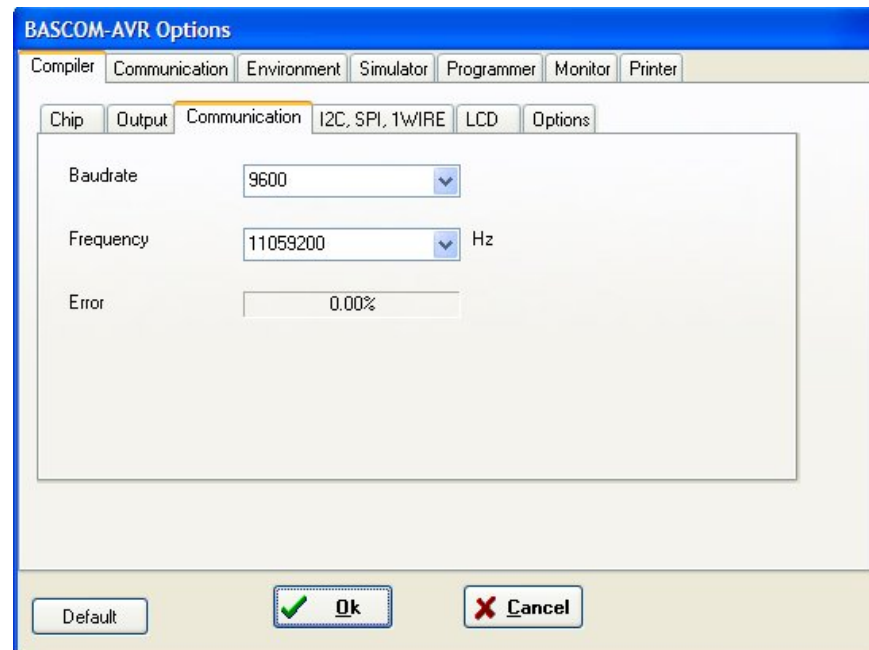
4. Perancangan Perangkat Lunak (*Software*)

a. Program

Sebelum menulis program tentunya ada beberapa pengaturan yang harus dilakukan. Bascom AVR *options* merupakan form pengaturan yang akan menyesuaikan antara program yang dibuat dengan mikrokontroler yang sebenarnya. Dengan menentukan pengaturan, maka programmer tidak perlu mendeklarasikan kembali nilai-nilai yang telah ditentukan. Salah satu pengaturan yang harus ditentukan adalah *compiler*.



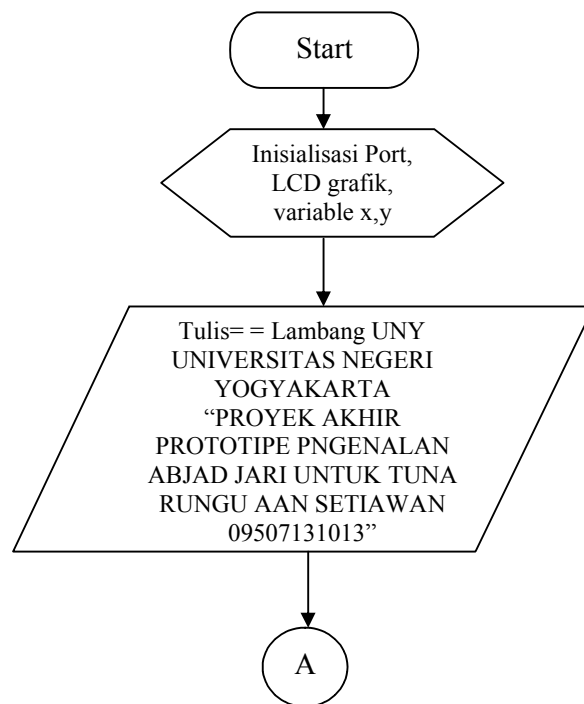
Gambar 19. Pengaturan Chip Pada Bascom AVR

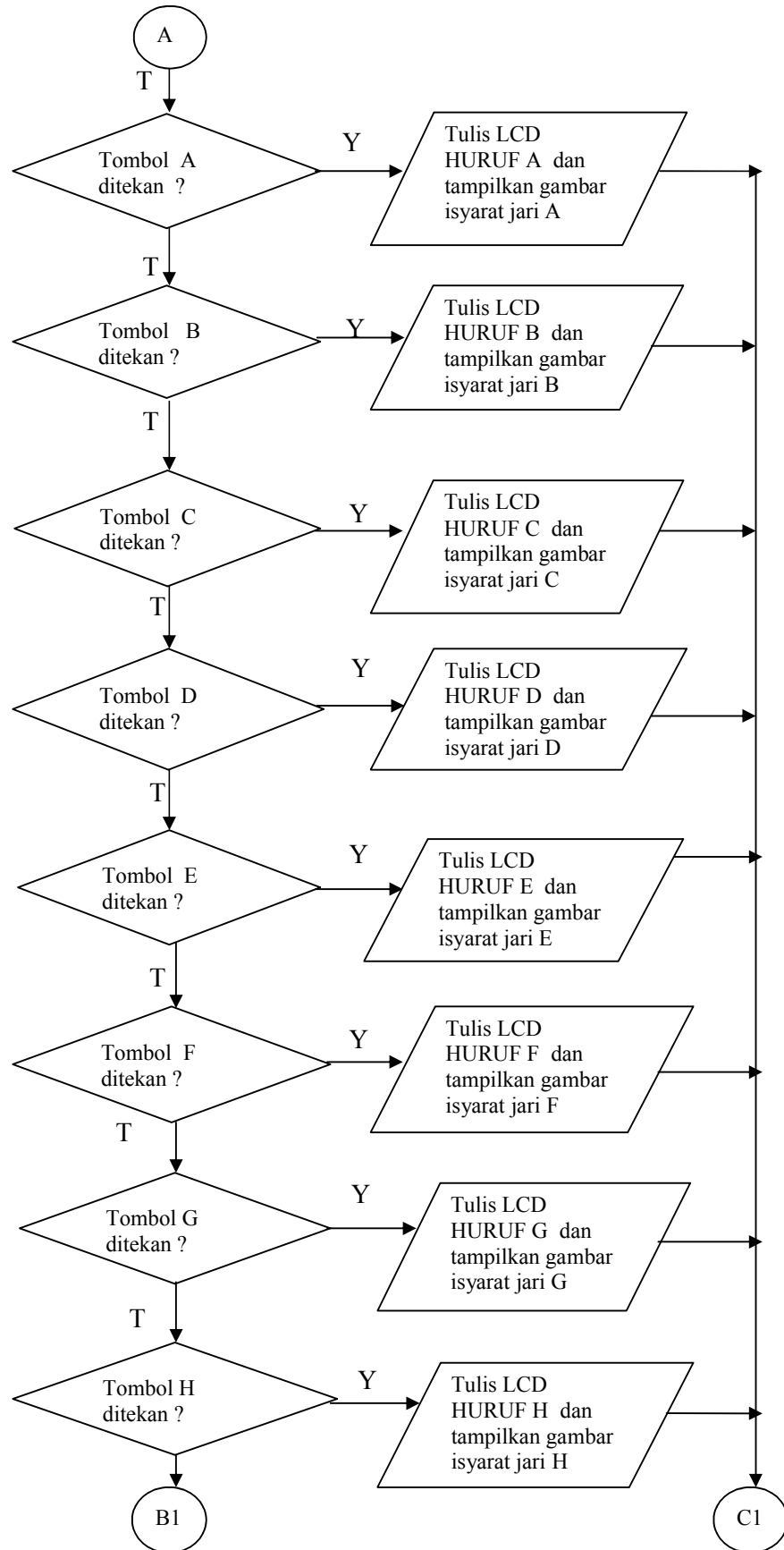


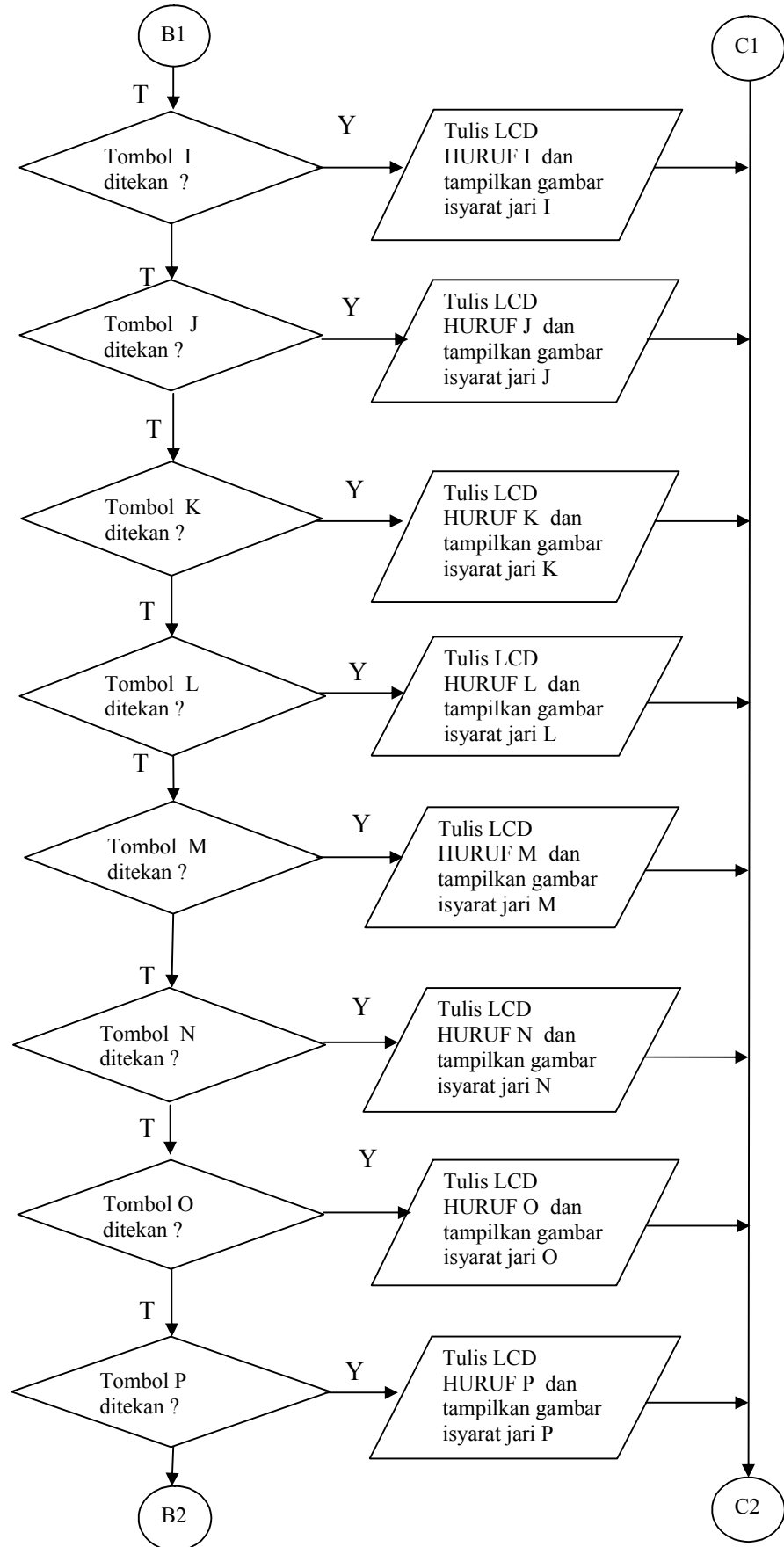
Gambar 20. Pengaturan *Communication* Pada Bascom AVR

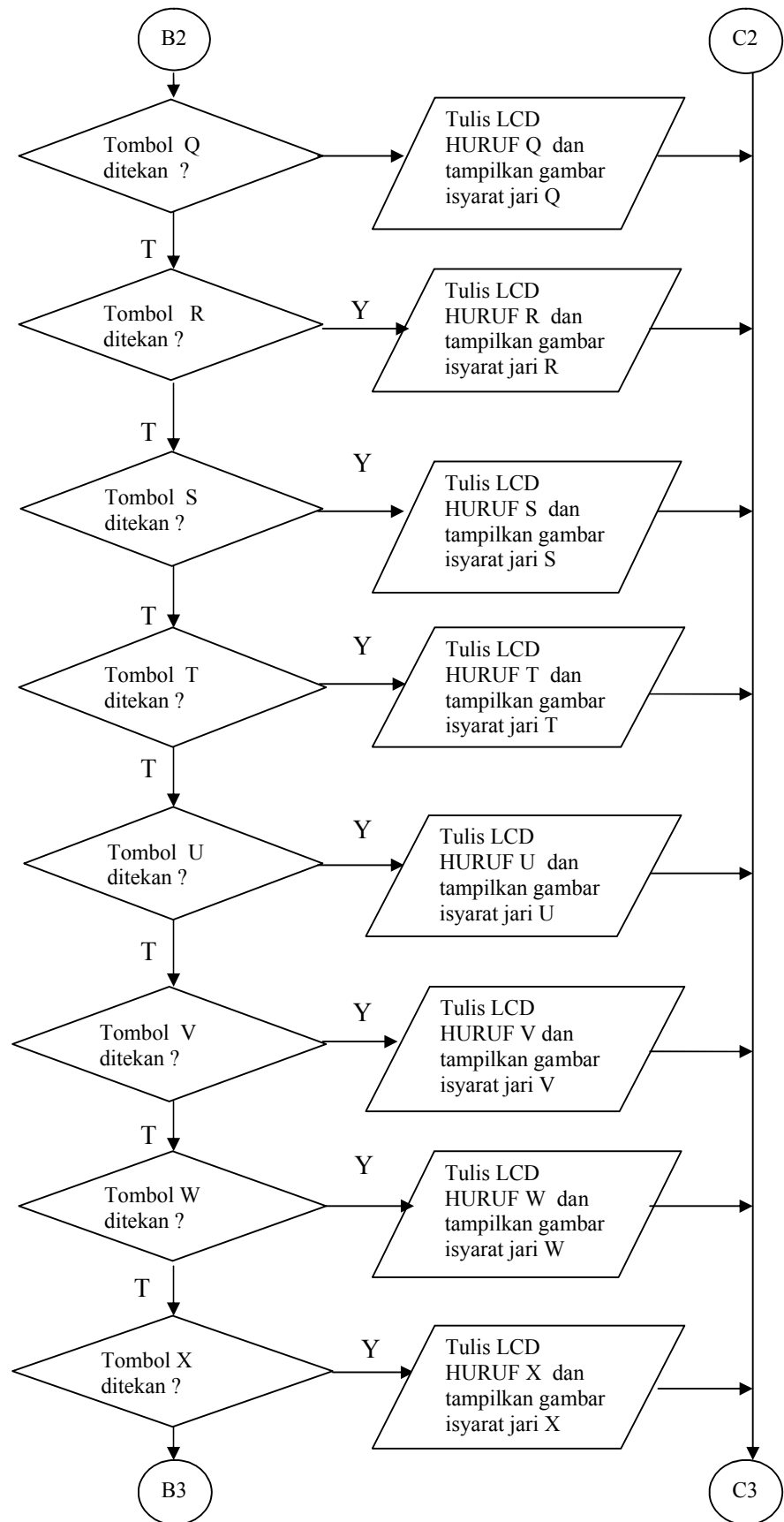
b. Flow chart

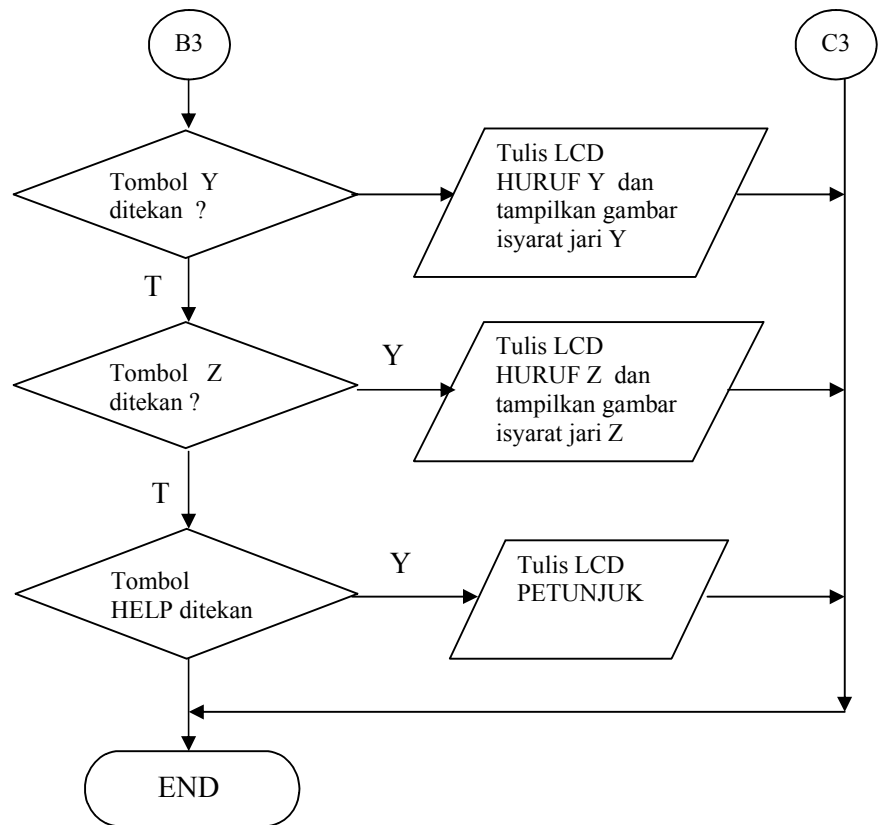
Flowchart program pada rangkaian pengenalan abjad jari untuk tuna rungu berbasis mikrokontroler ATmega 32 dimulai dari start yang berarti memulai program. Kemudian melakukan inisialisasi yang akan dibaca oleh program dan menjalankan program sudah dituliskan sampai akhir. Berikut ini *flowchart* program secara lengkap:











D. Pengoperasian Alat

Pengoperasian alat ini dapat dilakukan dengan cara sebagai berikut:

1. Menghubungkan ke tegangan 5 V dengan adaptor dan ic regulator 7805 lalu menekan tombol saklar untuk menghidupkan.
2. Menghubungkan keypad matriks dan LCD grafik ke port mikrokontroler.
3. Menekan keypad matriks sesuai dengan keadaan
4. Gambar tampilan dan tulisan pada *grafik LCD* sesuai dengan *input* yang diberikan melalui *keypad* .

BAB IV

PENGUJIAN DAN PEMBAHASAN

Tujuan dari pengujian dan pembahasan adalah untuk mengetahui kinerja alat baik secara per bagian blok rangkaian maupun sistem keseluruhan apakah sudah seperti yang diharapkan atau belum. Pengujian ini meliputi :

A. Hasil Pengujian

1. Pengujian Catu Daya

Pengujian catu daya dilakukan untuk mengetahui apakah catu daya yang dibuat sesuai yang diinginkan atau belum, dan untuk mengetahui tingkat kesalahan yang dihasilkan oleh IC regulator tersebut. Pengujian regulator dapat dilihat pada tabel dibawah ini :

Tabel 6. Pengujian Input dan Output Rangkaian Regulator

Regulator 5 V	
Input	Output
12V DC	5,2V DC

2. Pengujian Keypad Matriks 5x6

Rangkaian keypad berfungsi sebagai input yang berhubungan langsung dengan pengguna. Keypad disimpulkan bekerja dengan baik apabila hasil penekanan setiap tombol yang ada sesuai dengan program.

Tabel 7. Pengujian Penekanan Tombol Pada Keypad Matriks 5x6

Penekanan	Jalur Yang Aktif
Keypad A	Baris 1 (PA.0) – Kolom 1 (PC.0)
Keypad B	Baris 1 (PA.0) – Kolom 2 (PC.1)
Keypad C	Baris 1 (PA.0) – Kolom 3 (PC.2)
Keypad D	Baris 1 (PA.0) – Kolom 4 (PC.3)
Keypad E	Baris 1 (PA.0) – Kolom 5 (PC.4)

Lanjutan tabel.7 Pengujian penekanan tombol pada keypad Matriks 5x6

Penekanan	Jalur Yang Aktif
Keypad F	Baris 1 (PA.0) – Kolom 6 (PC.5)
Keypad G	Baris 2 (PA.1) – Kolom 1 (PC.0)
Keypad H	Baris 2 (PA.1) – Kolom 2 (PC.1)
Keypad I	Baris 2 (PA.1) – Kolom 3 (PC.2)
Keypad J	Baris 2 (PA.1) – Kolom 4 (PC.3)
Keypad K	Baris 2 (PA.1) – Kolom 5 (PC.4)
Keypad L	Baris 2 (PA.1) – Kolom 6 (PC.5)
Keypad M	Baris 3 (PA.2) – Kolom 1 (PC.0)
Keypad N	Baris 3 (PA.2) – Kolom 2 (PC.1)
Keypad O	Baris 3 (PA.2) – Kolom 3 (PC.2)
Keypad P	Baris 3 (PA.2) – Kolom 4 (PC.3)
Keypad Q	Baris 3 (PA.2) – Kolom 5 (PC.4)
Keypad R	Baris 3 (PA.2) – Kolom 6 (PC.5)
Keypad S	Baris 4 (PA.3) – Kolom 1 (PC.0)
Keypad T	Baris 4 (PA.3) – Kolom 2 (PC.1)
Keypad U	Baris 4 (PA.3) – Kolom 3 (PC.2)
Keypad V	Baris 4 (PA.3) – Kolom 4 (PC.3)
Keypad W	Baris 4 (PA.3) – Kolom 5 (PC.4)
Keypad X	Baris 4 (PA.3) – Kolom 6 (PC.5)
Keypad Y	Baris 5 (PA.4) – Kolom 3 (PC.2)
Keypad Z	Baris 5 (PA.4) – Kolom 4 (PC.3)
Keypad HELP	Baris 5 (PA.4) – Kolom 6 (PC.5)

3. Pengujian LCD Grafik

Pengujian sistem minimum mikrokontroler ATmega 32 dan rangkaian LCD menggunakan ISIS Profesional pada *software* Proteus 7.9 dengan membuat simulasi rangkaian keseluruhan. Rangkaian tersebut terdiri dari tiga komponen utama antara lain: catu daya, keypad matriks, ATmega 32 dan LCD grafik.

Pada saat pengujian, dalam rangkaian simulasi ATmega 32 diisikan program sederhana yang dibuat secara bertahap menggunakan *software* Bascom AVR 2.0.7.3. Apabila dalam simulasi sistem minimum mampu menampilkan tulisan dan karakter gambar pada LCD, maka sistem minimum dan rangkaian LCD telah bekerja sesuai dengan yang diharapkan. Selanjutnya dapat dibuat rangkaian dalam bentuk *hardware*.

Hasil pengujian sistem minimum pada rangkaian output LCD adalah sebagai berikut:




Gambar 21. Hasil pengujian sistem minimum dan LCD grafik.

4. Pengujian Seluruh Sistem

Tabel 8. Pengujian Sistem Secara Keseluruhan

Penekanan Tombol	Tampilan LCD grafik	Keterangan
Belum ditekan	Logo UNY	Selama 3 detik kemudian data di clearkan
Belum ditekan	UNIVERSITAS NEGERI YOGYAKARTA	“UNIVERSITAS”, selama 1 detik “NEGERI”, selama 1 detik “YOGYAKARTA”, selama 3 detik kemudian data clear

Lanjutan tabel 8. Pengujian sistem secara keseluruhan

Penekanan Tombol	Tampilan LCD grafik	Keterangan
Belum ditekan	PROYEK AKHIR PROTOTIPE PENGENALAN ABJAD JARI UNTUK TUNA RUNGU AAN SETIAWAN 09507131013	“PROYEK AKHIR”, selama 1 detik “PROTOTIPE PENGENALAN ABJAD JARI UNTUK TUNA RUNGU”, selama 1 detik “AAN SETIAWAN 09507131013”, selama 3 detik
Keypad A – Z ditekan secara bergantian		Tampilan setelah keypad “A” ditekan, sebelah kiri tampilan gambar isyarat jari dan sebelah kanan keterangan huruf abjad jari
Penekanan tombol HELP	Tampil secara bergantian beberapa petunjuk pemakaian alat.	Petunjuk penekanan tombol, cara memperagakan gambar dan cara mematikan alat dengan saklar ON/OFF.
Ditekan berulang kali	Gambar abjad jari akan tetap tampil seperti sebelumnya dan berkedip	Penekanan tombol yang sama
Ditekan bersamaan	Tampil salah satu abjad jari dari penekanan beberapa tombol	Penekanan sebanyak 2 tombol atau lebih secara bersamaan

B. Pembahasan

1. Hardware

a. Catu Daya

Berdasarkan hasil pengujian pada bagian rangkaian catu daya menunjukkan bahwa rangkaian ini telah bekerja dengan baik dimana tegangan yang masuk ke transformator sebesar 220 V_{AC} dan *output*

dari transformator sebelum melewati *rectifier* adalah $12 V_{AC}$. Setelah melalui *diode bridge* tegangannya menjadi $12 V_{DC}$. Agar menghasilkan tegangan yang stabil maka digunakan regulator tegangan $5 V_{DC}$, sehingga didapat *output* IC LM 7805 sebesar $5,2 V_{DC}$.

b. Keypad Matriks 5x6

Pada proyek akhir ini rangkaian keypad digunakan sebagai unit masukan. Masukan berupa penekanan keypad abjad melalui inisialisasi kode tombol matriks yang ditekan.

Proses inisialisasi tombol hingga sesuai yang diharapkan dapat dijelaskan pada halaman 10 pada bab 2. Apabila satu tombol pada keypad ditekan, maka akan terhubung antara jalur kolom (Port C) dengan baris (Port A). Dan hasil penekanan tersebut akan diinisialisasi sesuai program yang dijalankan.

c. LCD Grafik

Sesuai dengan pengujian yang telah dilakukan, sistem minimum mampu bekerja dengan baik. Hal tersebut ditunjukkan pada hasil tampilan LCD grafik sesuai dengan program yang diisikan. Itu semua ditunjukkan LCD grafik mampu menampilkan karakter-karakter yang diperintahkan oleh mikrokontroler diantaranya mampu menampilkan karakter gambar isyarat jari dan karakter huruf.

Untuk mengaplikasikan suatu gambar pada LCD Graphic dengan BASCOM AVR adalah dengan cara mengkonvert suatu gambar ke format *.bgf (Bascom Graphical File) yang sebelumnya berasal dari

file gambar yg berekstensi *.bmp menggunakan “**Graphic Converter**” (Tools => Graphic Converter)

2. Software

Perangkat lunak atau software pada alat ini dibuat menggunakan bahasa basic dengan Bascom AVR 2.0.7.3. Bagian program meliputi:

a. Definisi prosesor

```
$regfile = "m32def.dat"
```

Baris ini menyatakan bahwa chip yang digunakan adalah salah satu bagian keluarga AVR ATmega dengan seri 32.

b. Penyertaan fungsi

```
Dim Keypad As Byte , Tombol As Byte
```

```
Config Graphlcd = 128 * 64sed , Dataport = Portd  
,Controlport = Portb , Ce = 0 , Ce2 = 1 , Cd = 5 , Rd =  
6 , Reset = 4 , Enable = 7
```

Baris ini merupakan inialisasi pada port keypad dan LCD grafik.

d. Penyertaan Fungsi Utama

```
cls
```

```
Showpic 32 , 0 , Uny
```

```
wait 3
```

```
Uny:
```

```
$bgf "uny.bgf"
```

Perintah ini berfungsi untuk menampilkan karakter gambar.

```

cls
Setfont Font8x8 //pengaturan ukuran huruf

Lcdat 2 , 20 , "UNIVERSITAS" //menampilkan huruf

```

Perintah ini berfungsi untuk mengatur ukuran huruf dan menampilkan huruf di LCD grafik.

```

Do
Gosub Cek_tombol
Loop Until Keypad <> 255
waitms 20
Tombol = Keypad
Do
Gosub Cek_tombol
Loop Until Keypad = 255
Gosub Disply_huruf
Loop //perulangan yang digunakan untuk melakukan perulangan
      program keypad selama kondisi telah terpenuhi
End

```

Perintah ini berfungsi untuk mengatur penekanan keypad matriks.

```

Disply_huruf:
Select Case Tombol

Case 0 : cls

      showpic 0 , 0 , A

      Setfont Font8x8

      Lcdat 4 , 70 , "HURUF A"

.

.

```

Case 27 : Cls

Showpic 0 , 0 , z

SetFont Font8x8

Lcdat 4 , 70 , "HURUF Z"

Perintah ini adalah pengelompokkan data gambar dan huruf.

\$include "font8x8.font.bas"

Perintah ini berfungsi sebagai pemanggilan font untuk karakter huruf.

C. Unjuk Kerja

Unjuk kerja keseluruhan sistem ini merupakan kombinasi seluruh bagian *hardware* dan *software* yang telah terintegrasi menjadi sebuah sistem. Sesuai dengan hasil pengujian keseluruhan, maka sistem ini berfungsi dengan baik.

Unjuk kerja dari alat ini merupakan penerapan dari diagram alir program utama, maka dari itu untuk mengoperasikan alat ini harus berpedoman dari diagram alir program utama.

Ketika sistem tersebut dihidupkan, maka sistem siap bekerja. LCD menampilkan tampilan awal seperti hasil pengujian pada table 8. Selanjutnya instruksi terfokus pada keypad matriks sebagai input huruf abjad. Instruksi berupa kode keypad dengan fungsi setiap tombol. Selanjutnya LCD grafik akan menampilkan karakter gambar isyarat jari dan keterangan karakter huruf abjad jari.

Selain itu, penulis juga meminta masukan dari seorang pengajar siswa tuna rungu SLB Negeri 1 Wonosari tentang hasil alat yang telah dibuat. Masukan tentang alat yang dibuat dapat dilihat di lampiran pada halaman 51.

D. Pembahasan Masukan Pakar Tuna Rungu

Dari hasil pembuatan prototipe pengenalan abjad jari untuk tuna rungu, beliau mengucapkan terimakasih karena merasa terbantu dengan adanya mahasiswa yang peduli tentang anak berkebutuhan khusus. Selain itu beliau menjelaskan tahap awal cara mengajarkan ke anak tuna rungu tentang pengenalan huruf. Setelah anak mampu memahami huruf, kemudian anak diajarkan tentang bagaimana cara merangkai kata dasar dan kata imbuhan . Kata yang awal diperkenalkan kepada anak tuna rungu merupakan kata yang sering digunakan setiap hari seperti panggilan ke anggota keluarga, benda yang dipakai setiap hari dan kegiatan sehari-hari.

Cara berkomunikasi tuna rungu ada beberapa cara, antara lain: menggunakan gerak bibir, abjad jari, ekspresi dan gerakan tangan. Jadi, dari pengenalan huruf anak tuna rungu mampu mempelajari kata dasar dan merangkai sebuah kalimat. Selanjutnya anak tuna rungu dapat berkomunikasi dengan orang disekitarnya.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap “ *Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu Berbasis Mikrokontroller ATmega32* ” maka dapat disimpulkan :

1. Perangkat keras Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu Berbasis Mikrokontroller ATmega32 dapat diwujudkan dengan menggabungkan beberapa komponen dan rangkaian, diantaranya : rangkaian catu daya, keypad matriks, sistem minimum dan rangkaian *output* (LCD grafik). Setiap elemen tersebut disatukan oleh mikrokontroller ATmega 32 sebagai pusat kendali.
2. Perangkat lunak berupa program bahasa basic dibuat menggunakan Bascom AVR 2.0.7.3 , yang terdiri dari beberapa bagian: (1) Definisi *prosesor* (2) Penyertaan fungsi (3) Definisi *Port*, Deklarasi variabel (4) Fungsi Utama.meliputi *setting* awal pada port a sebagai *input* baris keypad matriks, port c sebagai *input* kolom keypad matriks, sedangkan port b dan port d sebagai *output* LCD grafik.
3. Unjuk kerja keseluruhan sistem ini merupakan kombinasi dari seluruh bagian *hardware* dan *software* yang telah terintegrasi menjadi sebuah sistem. Hasil pengujian yang dilakukan terhadap prototipe ini menunjukkan bahwa alat dapat bekerja dengan baik.

4. Pakar tuna rungu memberikan tanggapan bahwa prototipe ini sudah cukup baik. Tahap awal cara mengajarkan ke anak tuna rungu berkomunikasi adalah dengan pengenalan huruf abjad. Dari pengenalan huruf anak tuna rungu mampu mempelajari kata dasar dan merangkai sebuah kalimat. Selanjutnya anak tuna rungu dapat berkomunikasi dengan orang disekitarnya.

B. Saran

Dalam pembuatan Proyek Akhir ini terdapat kekurangan-kekurangan, sehingga diperlukan pengembangan lebih lanjut. Saran membangun yang dibutuhkan untuk menyempurnakan Proyek Akhir ini, antara lain sebagai berikut :

1. Menambah fungsi keypad untuk membentuk kata dasar, tampilan angka dan tanda baca.
2. Mengganti LCD grafik dengan ukuran dan pixel yang lebih besar, sehingga dapat menambah karakter gambar atau huruf yang lebih banyak dengan tampilan yang lebih jelas.
3. Alat ini masih menggunakan sumber listrik dari PLN ,dan untuk dapat menyuplai tegangan sendiri maka perlu dtambahkan baterai, sehingga saat listrik PLN padam, alat ini dapat menyuplai tegangan sendiri.
4. Dari pakar tuna rungu, beliau memberikan saran agar nantinya prototipe ini dapat dikembangkan lagi dengan pengenalan kata dasar dan kalimat, sehingga anak tuna rungu mampu belajar berkomunikasi dengan orang disekitarnya.

DAFTAR PUSTAKA

- Andreas Viklund. (2011). *Teori Keypad Matriks dan Cara Penggunaannya*.
<http://depokinstruments.com/2011/07/27/teori-keypad-matriks-4x4-dan-cara-penggunaannya/#more-1442> diakses tanggal 8 Oktober 2012
- Anggito Saputra. (2012). *Konsep Tuna Rungu*.
<http://anggitosaputra.blogspot.com/2012/06/konsep-tunarungu.html>
diakses tanggal 27 Februari 2013
- Atmel Corporation. 2003. *8-bit Microcontroller with 16K Bytes In-System Programmable Flash Atmega32 Atmega32L Preliminary*
[http:// www.alldatasheet.com](http://www.alldatasheet.com) diakses tanggal 10 Oktober 2012
- Digi-ware. (2012). *Graphic LCD 128x64 (C) /w blue STN white backlight*.
<http://www.digi-ware.com/img/d/WG12864CTMIVN.doc> diakses tanggal 26 Februari 2013
- Fahmizal. (2010). *Bascom AVR*
<https://fahmizaleeits.wordpress.com> diakses tanggal 02 Februari 2012
- Gunadi Sandjaja. (2006). *Tugas Akhir: Volume Unit Meter Dengan LCD Grafik*.
Surabaya : Universitas Kristen Petra.
- Hasanti Putra Gea. (2011). *Pendekatan Pengajaran Alternatif bagi Penyandang Tuna Rungu dan Tuna Wicara*.
<http://kwintal.blogspot.com/2011/05/27-pendekatan-pengajaran-alternatif.html> diakses tanggal 7 Oktober 2012
- Setiya Purnawan. (2009). *Tugas Akhir: Sistem Pemesanan Makanan Pada Restoran Modern Berbasis Mikrokontroler ATmega8535*. Yogyakarta : Universitas Negeri Yogyakarta
- Tim penyusun. 2011. *Pendoman proyek Akhir*. Fakultas Teknik UNY

LAMPIRAN

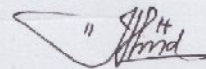
LEMBAR MASUKAN

Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu Berbasis
Mikrokontroler ATMEGA 32 .

No.	Bagian/ Materi	Masukan
1.	Keypad	<ul style="list-style-type: none">- Dapat dikembangkan lagi dengan tombol angka dan tanda baca.- Dapat ditambahkan dengan huruf kapital dan huruf kecil.
2.	Tampilan LCD	<ul style="list-style-type: none">- Dapat dikembangkan menggunakan LCD dengan ukuran dan pixel yang lebih besar, sehingga dapat menambah karakter gambar atau huruf yang lebih banyak dengan tampilan yang lebih jelas.
3.	Finishing	<ul style="list-style-type: none">- Dapat ditambahkan baterai untuk menyuplai tegangan sendiri.- Ketebalan casing dapat dikurangi menggunakan casing yang lebih tipis.

Yogyakarta, 15 Maret 2013

Pakar Tuna Rungu,



Siti Hunadah

NIP. 19590414 198303 2 007

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Siti Hunadah
Pekerjaan : Guru/Pengajar
Periode : Tahun 1983 - sekarang
Instansi : SLB NEGERI 1 WONOSARI

Menyatakan bahwa saya telah memberikan masukan dalam penyusunan proyek akhir berjudul "Prototipe Pengenalan Abjad Jari untuk Tuna Rungu" yang dilaksanakan oleh :

Nama : Aan Setiawan
NIM : 09507131013
Program Studi : Pendidikan Teknik Elektronika
Jurusan : Teknik Elektronika
Fakultas : Teknik

Demikian pernyataan saya, semoga masukan yang saya berikan dapat berguna untuk pengembangan proyek akhir ini.

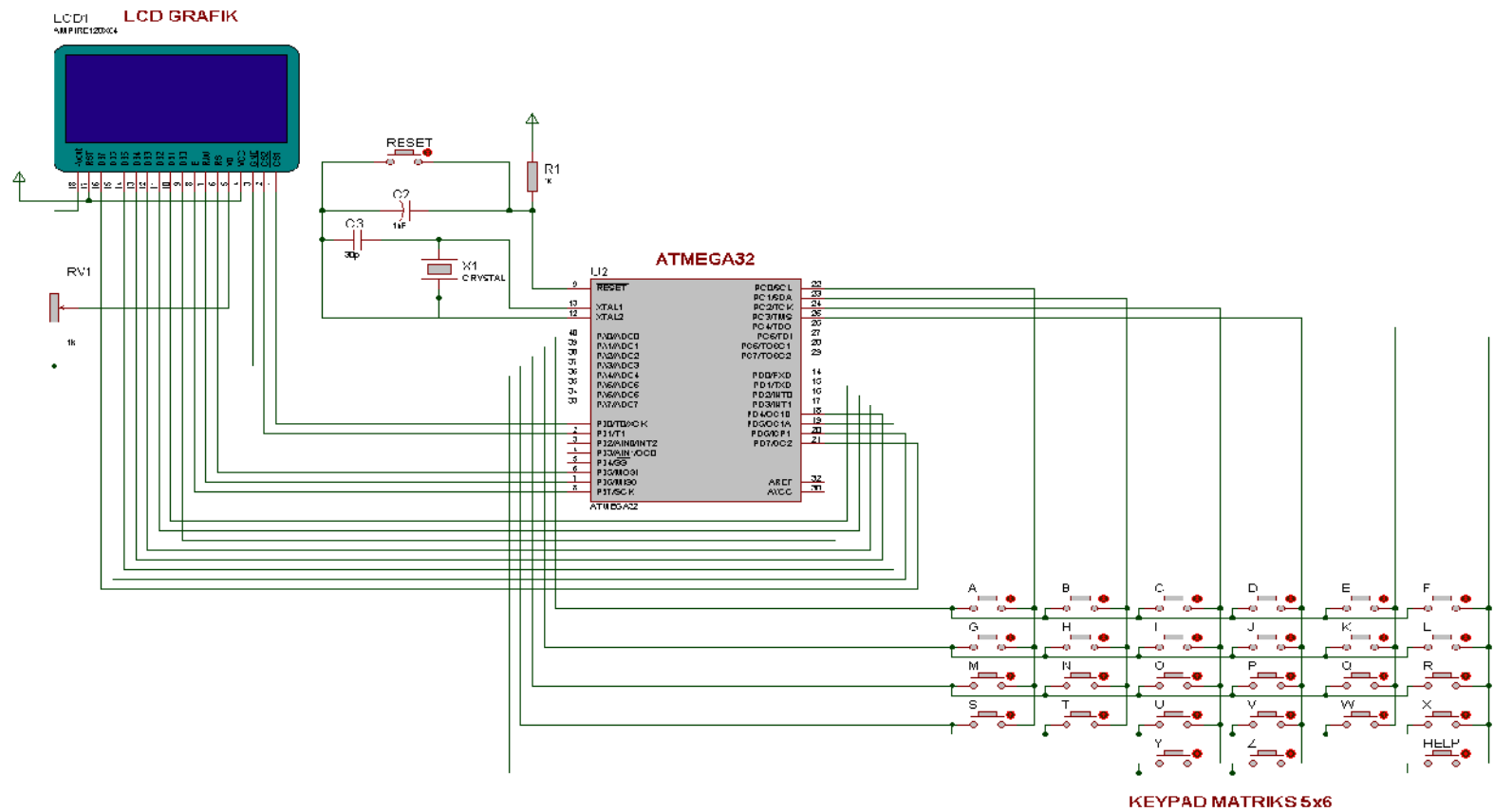
Yogyakarta, 15 Maret 2013

Pakar Tuna Rungu,

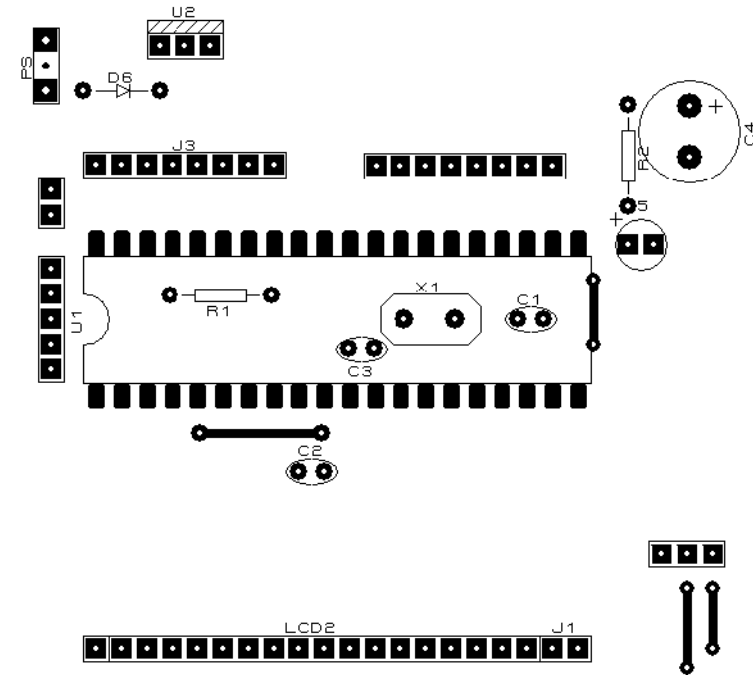
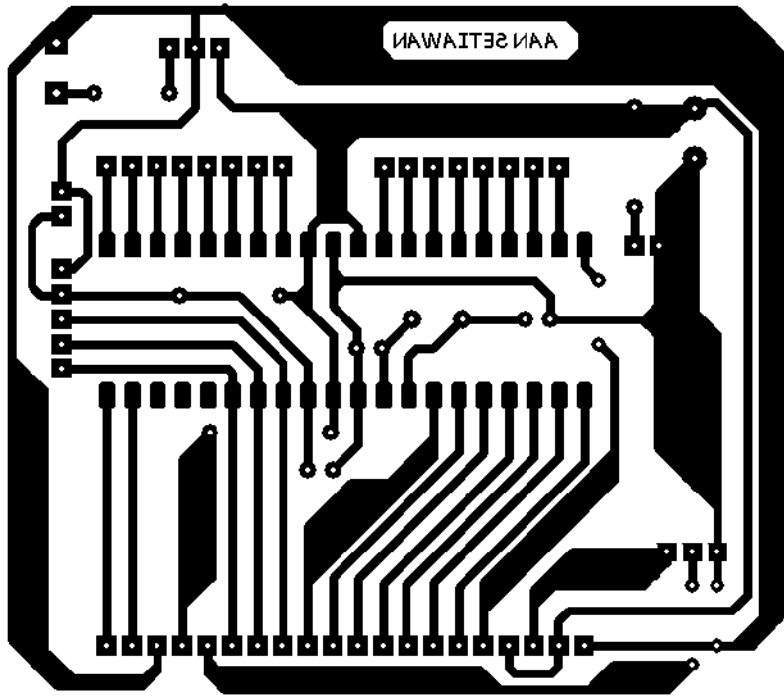


Siti Hunadah

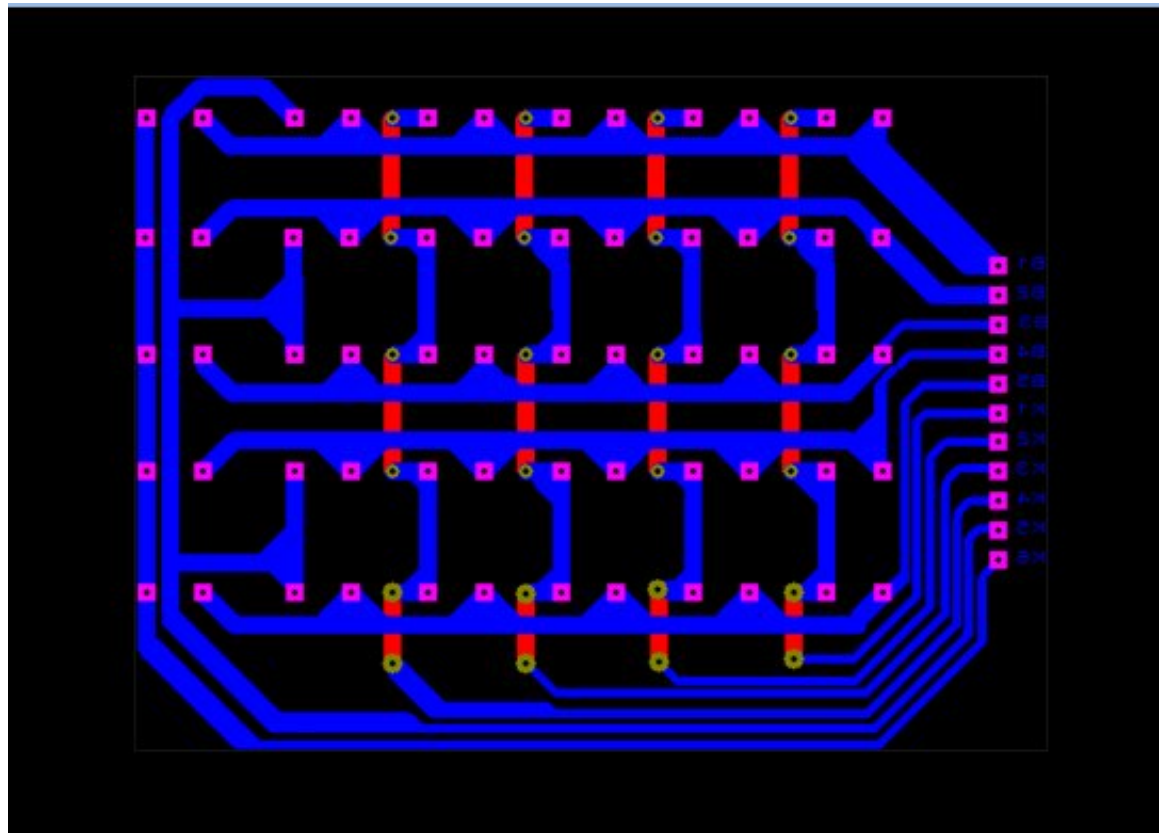
NIP. 19590414 198303 2 007



RANGKAIAN KESELURUHAN			KETERANGAN	
FT UNY	SKALA : 1 : 2	DIG: Aan S	A4	No.1
	DIP. Aris N	DIST. Aris N	NIM. 09507131013	



LAYOUT PCB RANGKAIAN			KETERANGAN	
			A4	No.2
FT UNY	SKALA : 1 : 1	DIG: Aan S		
	DIP. Aris N	DIST. Aris N		
			NIM. 09507131013	



LAYOUT KEYPAD MATRIKS 5x6			KETERANGAN	
			A4	No.3
FT UNY	SKALA : 1 ; 1	DIG: Aan S	NIM. 09507131013	
	DIP. Aris N	DIST. Aris N		

Graphic LCD 128x64 (C) /w blue STN white backlight

Contents

- 1.Module classification information
- 2.Precautions in Use of LCM
- 3.General Specification
- 4.Absolute Maximum Ratings
- 5.Electrical Characteristics
- 6.Optical Characteristics
- 7.Power Supply for LCD Module
- 8.Counter Drawing & Block Diagram
- 9.Interface Pin Function
- 10.Timing Characteristics
- 11.Display Control Instruction
- 12.Detailed Explanation
- 13.Quality Assurance
- 14.Reliability
- 15.Backlight Information

1. Module Classification Information

W G 1 2 8 6 4 C—T M I VN
 ① ② ③ ④ ⑤ ⑥ ⑦ ⑧

① Brand

② Display Type : H→Character Type, G→Graphic Type

③ Display Font : 128 * 64 Dots

④ Model serials number

⑤ Backlight Type :

N→Without backlight	T→LED, White
B→EL, Blue green	A→LED, Amber
D→EL, Green	R→LED, Red
W→EL, White	O→LED, Orange
F→CCFL, White	G→LED, Green
Y→LED, Yellow Green	

⑥ LCD Mode :

B→TN Positive, Gray	T→FSTN Negative
N→TN Negative,	
G→STN Positive, Gray	
Y→STN Positive, Yellow Green	
M→STN Negative, Blue	
F→FSTN Positive	

⑦ LCD Polarizer Type/ Temperature range/ View direction

A→Reflective, N.T, 6:00	H→Transflective, W.T, 6:00
D→Reflective, N.T, 12:00	K→Transflective, W.T, 12:00
G→Reflective, W. T, 6:00	C→Transmissive, N.T, 6:00
J→Reflective, W. T, 12:00	F→Transmissive, N.T, 12:00
B→Transflective, N.T, 6:00	I→Transmissive, W. T, 6:00
E→Transflective, N.T, 12:00	L→Transmissive, W.T, 12:00

⑧ Special Code V : Build in Negative Voltage N:NT0107 NT0108

2. Precautions in Use of LCD Module

- (1) Avoid applying excessive shocks to the module or making any alterations or modifications to it.
- (2) Don't make extra holes on the printed circuit board, modify its shape or change the components of LCD Module.
- (3) Don't disassemble the LCM.
- (4) Don't operate it above the absolute maximum rating.
- (5) Don't drop, bend or twist LCM.
- (6) Soldering: only to the I/O terminals.
- (7) Storage: please storage in anti-static electricity container and clean environment.

3. General Specification

ITEM	STANDARD VALUE	UNIT
Number of dots	128 ×64	dots
Outline dimension	78.0 (W) ×70.0 (H) ×14.3 (T)	mm
View area	62.0(W) ×44.0(H)	mm
Active area	56.3(W) ×38.38(H)	mm
Dot size	0.42(W) ×0.58(H)	mm
Dot pitch	0.44(W) ×0.60(H)	mm
LCD type	Transmissive , STN Negative, Blue	
View direction	6o'clock	
Backlight	LED , White	

4. Absolute Maximum Ratings

ITEM	SYMBOL	MIN.	TYP.	MAX.	UNIT
Operating Temperature	T_{OP}	-20	—	+70	°C
Storage Temperature	T_{ST}	-30	—	+80	°C
Input Voltage	V_I	0	—	V_{DD}	V
Supply Voltage For Logic	$V_{DD}-V_{SS}$	0	—	6.5	V
Supply Voltage For LCD	$V_{DD}-V_{LCD}$	0	—	17.0	V

5. Electrical Characteristics

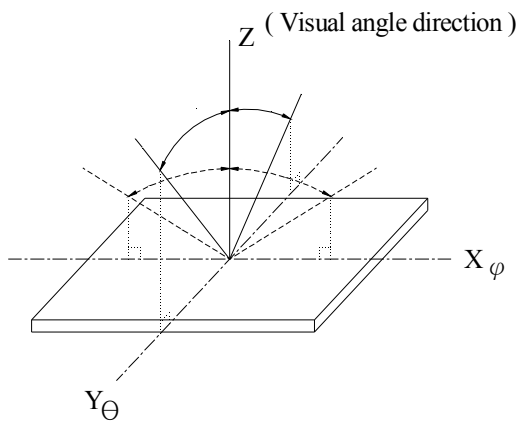
ITEM	SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT
Logic Voltage	$V_{DD}-V_{SS}$	—	4.5	5.0	5.5	V
Supply Voltage For LCD	$V_{DD}-V_{LCD}$	$T_a=-20^{\circ}\text{C}$	—	—	9.6	V
		$T_a=25^{\circ}\text{C}$	—	8.6	—	V
		$T_a=+70^{\circ}\text{C}$	8.2	—	—	V
Input High Volt.	V_{IH}	—	2.0	—	V_{DD}	V
Input Low Volt.	V_{IL}	—	0	—	0.8	V
Output High Volt.	V_{OH}	—	2.4	—	V_{DD}	V
Output Low Volt.	V_{OL}	—	0	—	0.4	V
Supply Current	I_{OP}	—	2.0	2.5	4.0	mA

6. Optical Characteristics

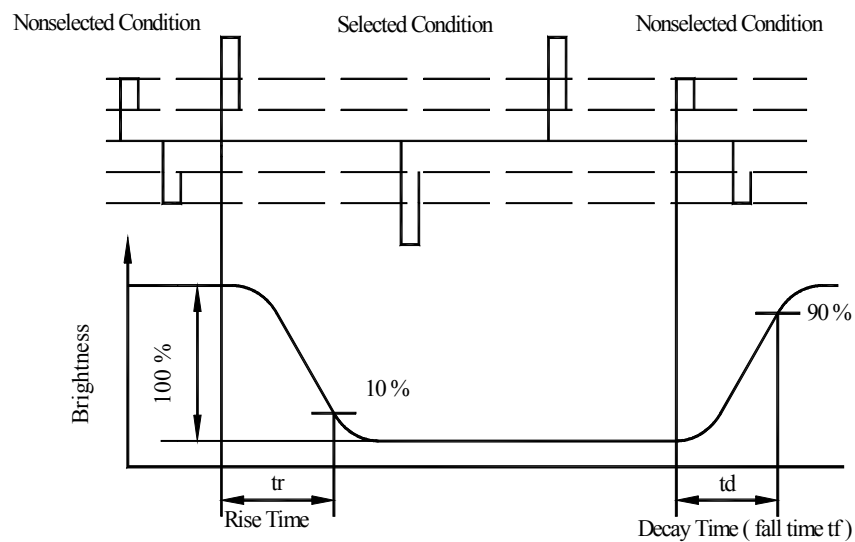
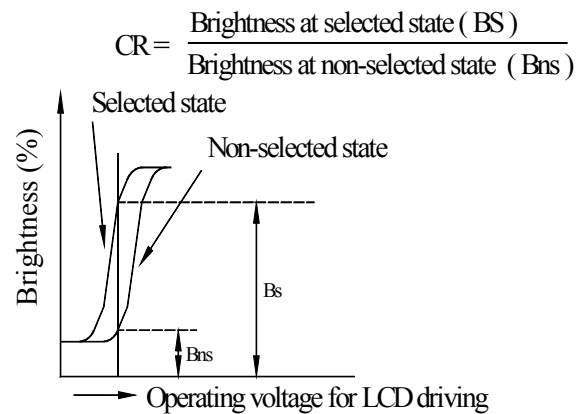
ITEM	SYMBAL	CONDITION	MIN	TYP	MAX	UNIT
View Angle	(V) θ	$CR \geq 2$	20	—	40	deg.
	(H) φ	$CR \geq 2$	-30	—	30	deg.
Contrast Ratio	CR	—	—	3	—	—
Response Time	T rise	—	—	200	300	ms
	T fall	—	—	150	200	ms

6.1 Definitions

■ View Angles

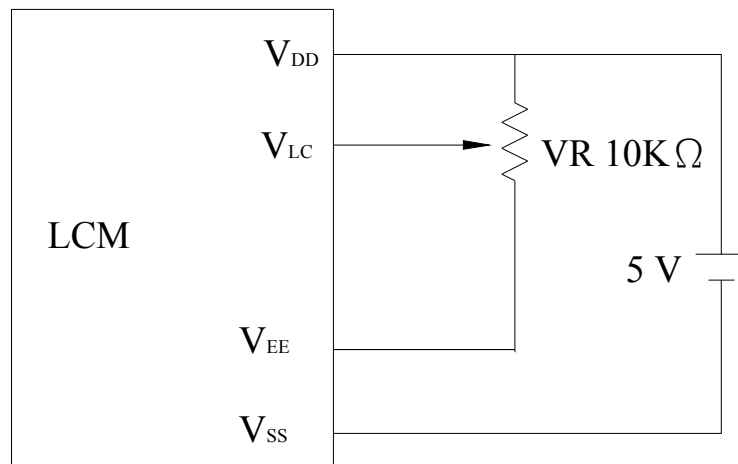


■ Contrast Ratio

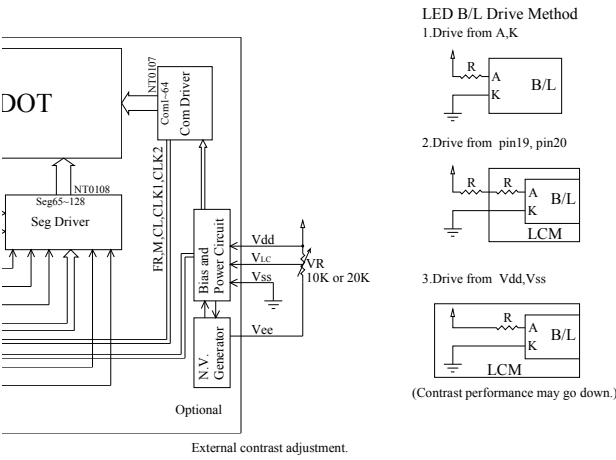
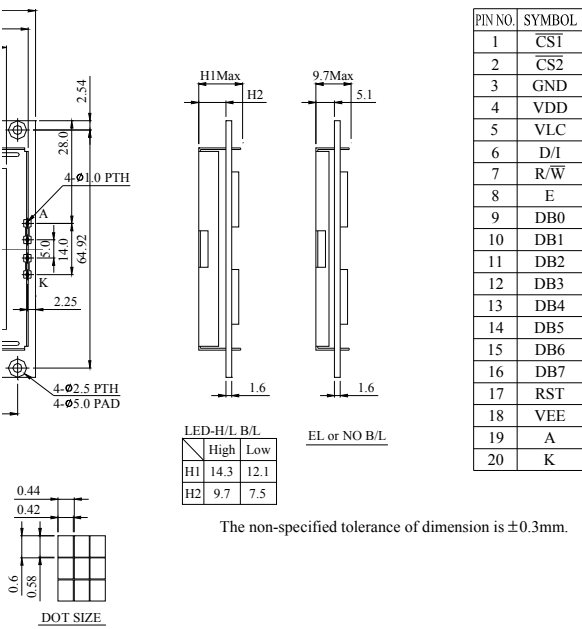


■ Response time

7. Power Supply for LCD Module



8.Counter Drawing & Block diagram



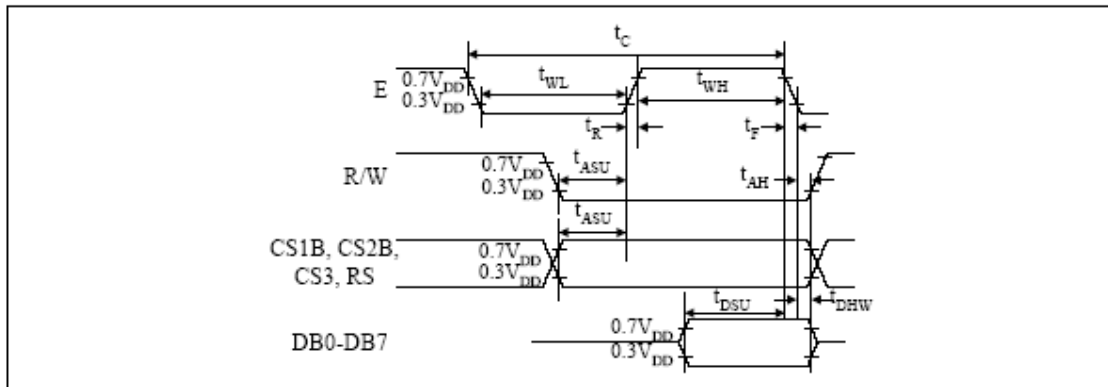
9. Interface Pin Function

Pin No.	Symbol	Level	Description
1	$\overline{\text{CS1}}$	L	Select Segment 1 ~ Segment 64
2	$\overline{\text{CS2}}$	L	Select Segment 65 ~ Segment128
3	GND	0V	Ground
4	V _{DD}	5.0V	Supply voltage for logic
5	V _{LC} /VO	(Variable)	Operating voltage for LCD
6	D/I	H/L	H: Data , L: Instruction
7	R/W	H/L	H: Read(MPU←Module) , L :Write(MPU→Module)
8	E	H	Enable signal
9	DB0	H/L	Data bit 0
10	DB1	H/L	Data bit 1
11	DB2	H/L	Data bit 2
12	DB3	H/L	Data bit 3
13	DB4	H/L	Data bit 4
14	DB5	H/L	Data bit 5
15	DB6	H/L	Data bit 6
16	DB7	H/L	Data bit 7
17	RST	L	Reset the LCM
18	VEE	V	Negative voltage;
19	A	—	Power supply for LED +
20	K	—	Power supply for LED -

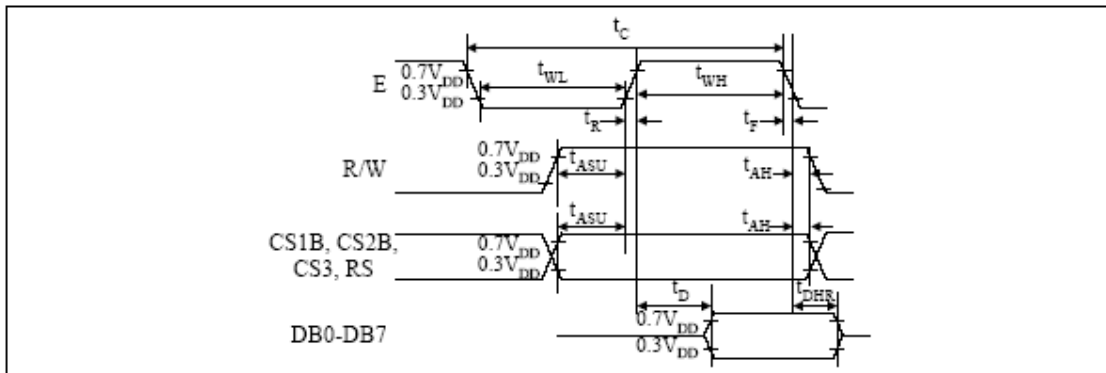
10. Timing Characteristics

MPU Interface

Characteristic	Symbol	Min	Typ	Max	Unit
E cycle	t _{cyc}	1000	—	—	ns
E high level width	t _{whE}	450	—	—	ns
E low level width	t _{wlE}	450	—	—	ns
E rise time	t _r	—	—	25	ns
E fall time	t _f	—	—	25	ns
Address set-up time	t _{as}	140	—	—	ns
Address hold time	t _{ah}	10	—	—	ns
Data set-up time	t _{dsu}	200	—	—	ns
Data delay time	t _{ddr}	—	—	320	ns
Data hold time (write)	t _{dhw}	10	—	—	ns
Data hold time (read)	t _{dhr}	20	—	—	ns



MPU Write Timing



MPU Read Timing

11. Display Control Instruction

The display control instructions control the internal state of the NT7108. Instruction is received from MPU to NT7108 for the display control. The following table shows various instructions.

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display on/off	L	L	L	L	H	H	H	H	H	L/H	Controls the display on or off. Internal status and display RAM data is not affected. L: OFF, H: ON
Set address (Y address)	L	L	L	H	Y address (0-63)						Sets the Y address in the Y address counter.
Set page (X address)	L	L	H	L	H	H	H	Page (0-7)			Sets the X address at the X address register.
Display Start line (Z address)	L	L	H	H	Display start line (0-63)						Indicates the display data RAM displayed at the top of the screen.
Status read	L	H	Busy	L	On/ Off	Reset	L	L	L	L	Read status. BUSY L: Ready H: In operation ON/OFF L: Display ON H: Display OFF RESET L: Normal H: Reset
Write display data	H	L	Write data								Writes data (DB0: 7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read display data	H	H	Read data								Reads data (DB0: 7) from display data RAM to the data bus.

12. Detailed Explanation

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	D

The display data appears when D is 1 and disappears when D is 0. Though the data is not on the screen with D=0, it remains in the display data RAM. Therefore, you can make it appear by changing D=0 into D=1.

SET ADDRESS (Y ADDRESS)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Y address (AC0-AC5) of the display data RAM is set in the Y address counter. An address is set by instruction and increased by 1 automatically by read or write operations of display data.

SET PAGE (X ADDRESS)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	AC2	AC1	AC0

X address (AC0-AC2) of the display data RAM is set in the X address register. Writing or reading to or from MPU is executed in this specified page until the next page is set.

DISPLAY START LINE (Z ADDRESS)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	AC5	AC4	AC3	AC2	AC1	AC0

Z address (AC0-AC5) of the display data RAM is set in the display start line register and displayed

at the top of the screen. When the display duty cycle is 1/64 or others (1/32-1/64), the data of total line number of LCD screen, from the line specified by display start line instruction, is displayed.

STATUS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BUSY	0	ON/OFF	RESET	0	0	0	0

- BUSY

When BUSY is 1, the Chip is executing internal operation and no instructions are accepted.

When BUSY is 0, the Chip is ready to accept any instructions.

- ON/OFF

When ON/OFF is 1, the display is OFF.

When ON/OFF is 0, the display is ON.

- RESET

When RESET is 1, the system is being initialized.

In this condition, no instructions except status read can be accepted.

When RESET is 0, initializing has finished and the system is in usual operation condition.

WRITE DISPLAY DATA

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Writes data (D0-D7) into the display data RAM. After writing instruction, Y address is increased by 1 automatically.

READ DISPLAY DATA

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Reads data (D0-D7) from the display data RAM. After reading instruction, Y address is increased by 1 automatically.

13. Quality Assurance

◆ Screen Cosmetic Criteria

No.	Defect	Judgement Criterion	Partition																				
1	Spots	A)Clear <table><tr><th>Size:d mm</th><th>Acceptable Qty in active area</th></tr><tr><td>$d \leq 0.1$</td><td>Disregard</td></tr><tr><td>$0.1 < d \leq 0.2$</td><td>6</td></tr><tr><td>$0.2 < d \leq 0.3$</td><td>2</td></tr><tr><td>$0.3 < d$</td><td>0</td></tr></table> Note: Including pin holes and defective dots which must be within one pixel size. B)Unclear <table><tr><th>Size:d mm</th><th>Acceptable Qty in active area</th></tr><tr><td>$d \leq 0.2$</td><td>Disregard</td></tr><tr><td>$0.2 < d \leq 0.5$</td><td>6</td></tr><tr><td>$0.5 < d \leq 0.7$</td><td>2</td></tr><tr><td>$0.7 < d$</td><td>0</td></tr></table>	Size:d mm	Acceptable Qty in active area	$d \leq 0.1$	Disregard	$0.1 < d \leq 0.2$	6	$0.2 < d \leq 0.3$	2	$0.3 < d$	0	Size:d mm	Acceptable Qty in active area	$d \leq 0.2$	Disregard	$0.2 < d \leq 0.5$	6	$0.5 < d \leq 0.7$	2	$0.7 < d$	0	Minor
Size:d mm	Acceptable Qty in active area																						
$d \leq 0.1$	Disregard																						
$0.1 < d \leq 0.2$	6																						
$0.2 < d \leq 0.3$	2																						
$0.3 < d$	0																						
Size:d mm	Acceptable Qty in active area																						
$d \leq 0.2$	Disregard																						
$0.2 < d \leq 0.5$	6																						
$0.5 < d \leq 0.7$	2																						
$0.7 < d$	0																						
2	Bubbles in Polarizer	<table><tr><th>Size:d mm</th><th>Acceptable Qty in active area</th></tr><tr><td>$d \leq 0.3$</td><td>Disregard</td></tr><tr><td>$0.3 < d \leq 1.0$</td><td>3</td></tr><tr><td>$1.0 < d \leq 1.5$</td><td>1</td></tr><tr><td>$1.5 < d$</td><td>0</td></tr></table>	Size:d mm	Acceptable Qty in active area	$d \leq 0.3$	Disregard	$0.3 < d \leq 1.0$	3	$1.0 < d \leq 1.5$	1	$1.5 < d$	0	Minor										
Size:d mm	Acceptable Qty in active area																						
$d \leq 0.3$	Disregard																						
$0.3 < d \leq 1.0$	3																						
$1.0 < d \leq 1.5$	1																						
$1.5 < d$	0																						
3	Scratch	In accordance with spots cosmetic criteria. When the light reflects on the panel surface, the scratches are not to be remarkable.	Minor																				
4	Allowable Density	Above defects should be separated more than 30mm each other.	Minor																				
5	Coloration	Not to be noticeable coloration in the viewing area of the LCD panels. Back-light type should be judged with back-light on	Minor																				

		state only.	
--	--	-------------	--

14. RELIABILITY

■ Content of Reliability Test

Environmental Test			
Test Item	Content of Test	Test Condition	Applicable Standard
High Temperature storage	Endurance test applying the high storage temperature for a long time.	80℃ 200hrs	—
Low Temperature storage	Endurance test applying the high storage temperature for a long time.	-30℃ 200hrs	—
High Temperature Operation	Endurance test applying the electric stress (Voltage & Current) and the thermal stress to the element for a long time.	70℃ 200hrs	—
Low Temperature Operation	Endurance test applying the electric stress under low temperature for a long time.	-20℃ 200hrs	—
High Temperature Humidity Storage	Endurance test applying the high temperature and high humidity storage for a long time.	80℃,90%RH 96hrs	—
High Temperature Humidity Operation	Endurance test applying the electric stress (Voltage & Current) and temperature / humidity stress to the element for a long time.	70℃,90%RH 96hrs	—
Temperature Cycle	Endurance test applying the low and high temperature cycle. <div style="text-align: center;"> <p>-30℃ 25℃ 80℃</p> <p>30min 5min 30min 1 cycle</p> </div>	-30℃/80℃ 10 cycles	—
Mechanical Test			
Vibration test	Endurance test applying the vibration during transportation and using.	Total fixed amplitude : 15mm Vibration Frequency : 10~55Hz One cycle 60 seconds to 3 directions of X,Y,Zfor Each 15 minutes	—

Others			
Static electricity test	Endurance test applying the electric stress to the terminal.	VS=800V,RS=1.5k Ω CS=100Pf 1 time	—

***Supply voltage for logic system=5V. Supply voltage for LCD system =Operating voltage at 25°C

15.Backlight Information

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	TEST CONDITION
Supply Current	I _{LED}	65	80	100	mA	V=3.5V
Supply Voltage	V	3.4	3.5	3.6	V	
Reverse Voltage	V _R	—	—	5	V	
Luminous Intensity	I _V	150	190	—	CD/M ²	I _{LED} =80mA
Life Time		—	50K	—	Hr.	I _{LED} ≤80mA
Color	White					

Note: The LED of B/L is drive by current only, drive voltage is for reference only.

drive voltage can make driving current under safety area (current between minimum and maximum).

DATA SHEET ATMEGA 32

Features

- High-performance, Low-power Atmel®AVR®8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 32Kbytes of In-System Self-programmable Flash program memory
 - 1024bytes EEPROM
 - 2Kbytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/1,000,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit A/D
 - 3 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, A/D Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega32L
 - 4.5V - 5.5V for ATmega32
- Speed Grades
 - 0 - 8MHz for ATmega32L
 - 0 - 16MHz for ATmega32
- Power Consumption at 1MHz, 3V, 25°C
 - Active: 1.1mA
 - Idle Mode: 0.25mA
 - Power-down Mode: < 1µA



8-bit AVR[®]
Microcontroller
with 32KBytes
In-System
Programmable
Flash

ATmega32
ATmega32L

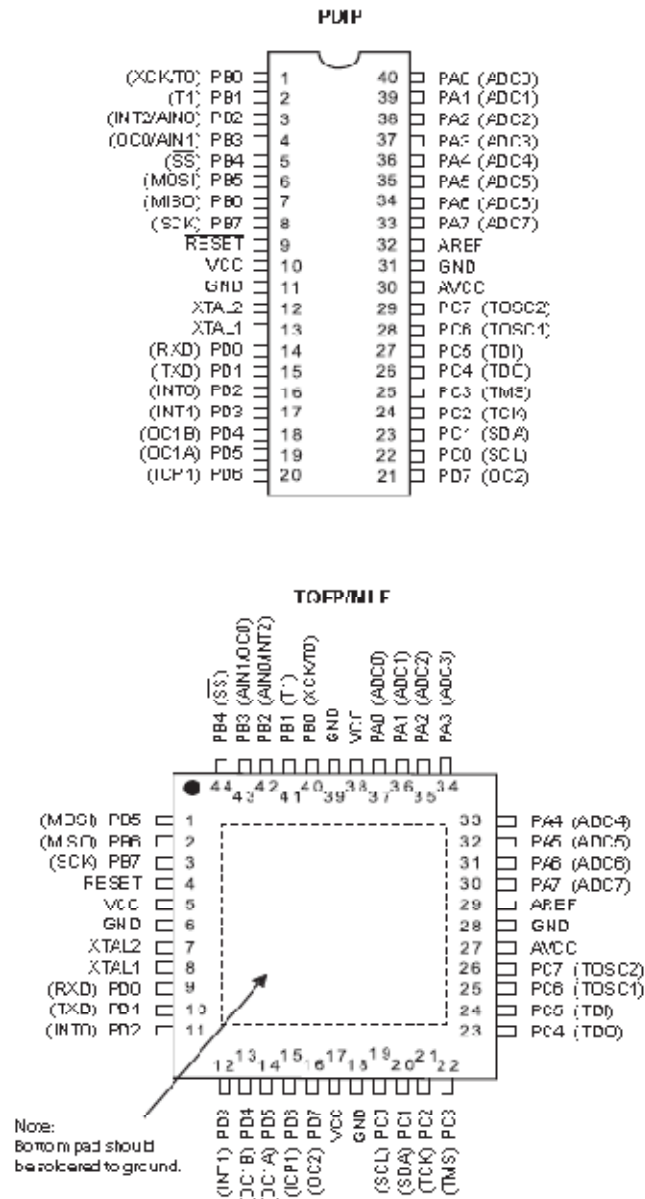
2532U-AVR-H-02/11



ATmega32(L)

Pin Configurations

Figure 1. Pinout ATmega32



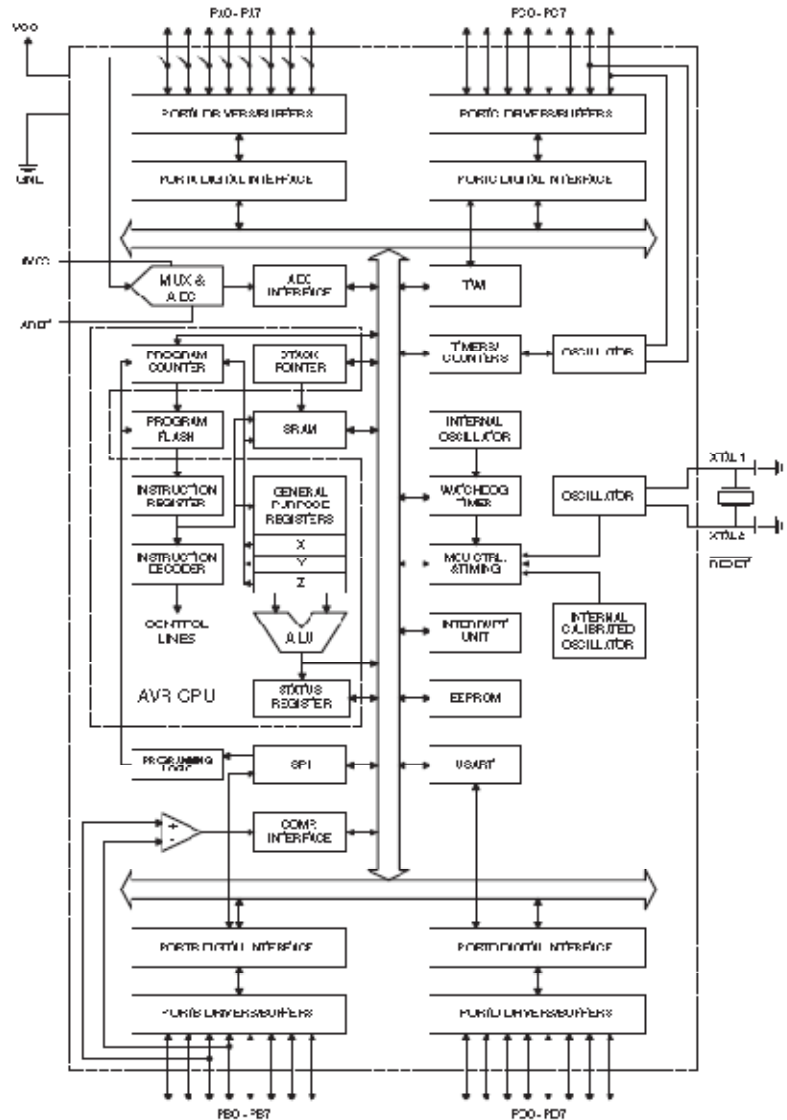
ATmega32(L)

Overview

The Atmel® AVR® ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



ATmega32(L)

The Atmel®AVR®AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega32 is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	<p>Port A serves as the analog inputs to the A/D Converter.</p> <p>Port A also serves as an 8-bit bidirectional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current; if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>

ATmega32(L)

Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega32 as listed on page 57.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bidirectional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>The TDO pin is tri-stated unless TAP states that shift out data are entered.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on page 60.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bidirectional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega32 as listed on page 62.</p>
RESET	<p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 16 on page 37. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V _{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V _{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.

ATmega32(L)

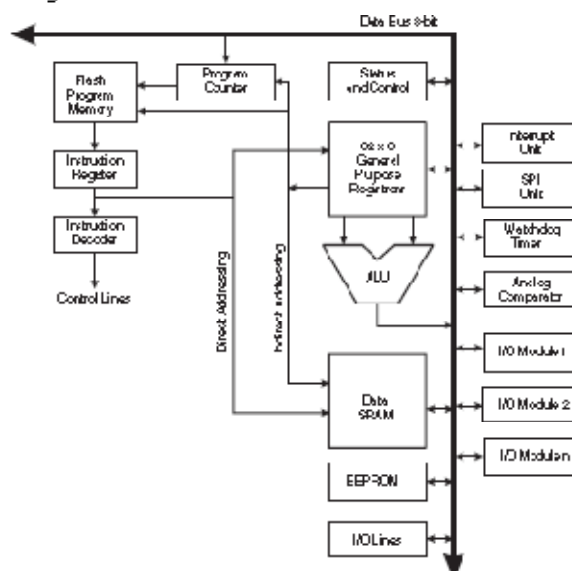
AVR CPU Core

Introduction

This section discusses the *Atmel*®AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single-level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is 16-Kbit System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for Look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

ATmega32(L)

Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SFI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, \$20 - \$5F.

ALU – Arithmetic Logic Unit

The high-performance Atmel®AVR® ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

ATmega32(L)

ATmega32 Memories

This section describes the different memories in the Atmel® AVR® ATmega32. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega32 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

In-System Reprogrammable Flash Program Memory

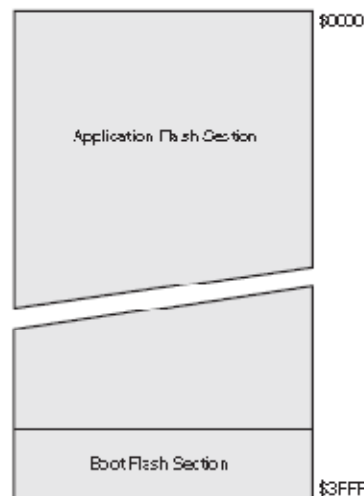
The ATmega32 contains 32 Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 16K × 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega32 Program Counter (PC) is 14 bits wide, thus addressing the 16K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in ["Boot Loader Support – Read-While-Write Self-Programming"](#) on page 244. ["Memory Programming"](#) on page 256 contains a detailed description of Flash Programming in SPI, JTAG or Parallel Programming mode.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory Instruction Description).

Timing diagrams for instruction fetch and execution are presented in ["Instruction Execution Timing"](#) on page 13.

Figure 8. Program Memory Map



ATmega32(L)

SRAM Data Memory

Figure 9 shows how the Atmel®AVR®ATmega32 SRAM Memory is organized.

The lower 2144 Data Memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 2048 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect Addressing Pointer Registers.

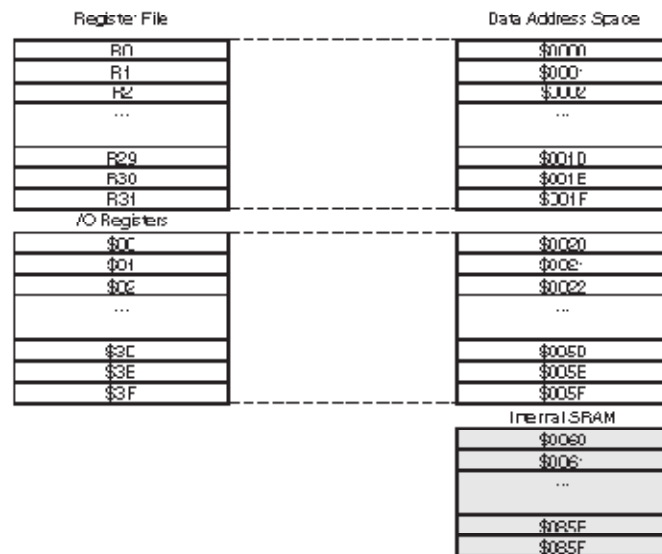
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 60 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 2048 bytes of internal data SRAM in the ATmega32 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 11.

Figure 9. Data Memory Map



Program Utama

'Aan Setiawan

'Prototipe Pengenalan Abjad Jari Untuk Tuna Rungu

\$lib "glcdKS108.lib"

\$crystal = 12000000

\$regfile = "m32def.dat"

\$hwstack = 100

\$swstack = 100

\$framesize = 0

Dim Keypad As Byte , Tombol As Byte

Config Graphlcd = 128 * 64sed , Dataport = Portd , Controlport =
Portb , Ce = 0 , Ce2 = 1 , Cd = 5 , Rd = 6 , Reset = 4 , Enable =
7

Config Porta = Output

Config Portc = Input

Porta = 255

Portc = 255

Kolom1 Alias Pinc.0

Kolom2 Alias Pinc.1

Kolom3 Alias Pinc.2

Kolom4 Alias Pinc.3

Kolom5 Alias Pinc.4

Kolom6 Alias Pinc.5

Baris1 Alias Porta.0

Baris2 Alias Porta.1

Baris3 Alias Porta.2

Baris4 Alias Porta.3

Baris5 Alias Porta.4

Const Bam = 0

```

Cls
Showpic 32 , 0 , Uny
wait 3
Cls
SetFont Font8x8
Lcdat 2 , 20 , "UNIVERSITAS"
wait 1
Lcdat 4 , 40 , "NEGERI"
wait 1
Lcdat 6 , 24 , "YOGYAKARTA"
wait 3
Cls
SetFont Font8x8
Lcdat 1 , 15 , "PROYEK AKHIR"
wait 1
Lcdat 3 , 25 , "PROTOTIPE"
Lcdat 4 , 20 , "PENGENALAN"
Lcdat 5 , 20 , "ABJAD JARI"
Lcdat 6 , 1 , "UNTUK TUNA RUNGU"
wait 1
Lcdat 7 , 15 , "AAN SETIAWAN"
Lcdat 8 , 20 , "09507131013"
wait 3
Cls
SetFont Font 8x8
Lcdat 1 , 30 , "PETUNJUK"
Lcdat 3 , 1 , "Untuk memulai,"
Lcdat 4 , 1 , "tekan tombol"
Lcdat 5 , 1 , "A sampai Z "
Lcdat 6 , 1 , "secara"
Lcdat 7 , 1 , "bergantian"
wait 4

```

Cls

Lcdat 1 , 30 , "PETUNJUK"

Lcdat 3 , 1 , "Gambar"

Lcdat 4 , 1 , "diperagakan"

Lcdat 5 , 1 , "dengan "

Lcdat 6 , 1 , "posisi tangan"

Lcdat 7 , 1 , "menghadap"

Lcdat 8 , 1 , "ke depan"

wait 4

Cls

Lcdat 1 , 30 , "PETUNJUK"

Lcdat 3 , 1 , "Gambar"

Lcdat 4 , 1 , "diperagakan"

Lcdat 5 , 1 , "dengan "

Lcdat 6 , 1 , "gerakan"

Lcdat 7 , 1 , "menurut"

Lcdat 8 , 1 , "anak panah"

wait 4

Cls

Lcdat 1 , 30 , "PETUNJUK"

Lcdat 3 , 1 , "Untuk bantuan"

Lcdat 4 , 1 , "tekan tombol"

Lcdat 5 , 35 , "HELP"

wait 4

Cls

Lcdat 1 , 30 , "PETUNJUK"

Lcdat 3 , 1 , "Setelah selesai,"

Lcdat 4 , 1 , "matikan"

Lcdat 5 , 1 , "saklar ON/OFF"

Lcdat 6 , 1 , "dibagian bawah"

Lcdat 7 , 1 , "dan lepas"

Lcdat 8 , 1 , "kabel power"

```

wait 4
Cls
Lcdat 4 , 5 , "SELAMAT BELAJAR"
wait 1
Do
    Do
        Gosub Cek_tombol
        Loop Until Keypad <> 255
        waitms 20
        Tombol = Keypad
    Do
        Gosub Cek_tombol
        Loop Until Keypad = 255
    Gosub Disply_huruf
Loop
End

```

```

Cek_tombol:
Porta = 255
Portc = 255
Keypad = 255
Baris1 = 0
Baris2 = 1
Baris3 = 1
Baris4 = 1
Baris5 = 1
If Kolom1 = 0 Then
    Keypad = 0
    Return
End If
If Kolom2 = 0 Then
    Keypad = 1

```

```
        Return
    End If
    If Kolom3 = 0 Then
        Keypad = 2
        Return
    End If
    If Kolom4 = 0 Then
        Keypad = 3
        Return
    End If
    If Kolom5 = 0 Then
        Keypad = 4
        Return
    End If
    If Kolom6 = 0 Then
        Keypad = 5
        Return
    End If
    Baris1 = 1
    Baris2 = 0
    Baris3 = 1
    Baris4 = 1
    Baris5 = 1
    If Kolom1 = 0 Then
        Keypad = 6
        Return
    End If
    If Kolom2 = 0 Then
        Keypad = 7
        Return
    End If
    If Kolom3 = 0 Then
```

```
        Keypad = 8
        Return
    End If
    If Kolom4 = 0 Then
        Keypad = 9
        Return
    End If
    If Kolom5 = 0 Then
        Keypad = 10
        Return
    End If
    If Kolom6 = 0 Then
        Keypad = 11
        Return
    End If
    Baris1 = 1
    Baris2 = 1
    Baris3 = 0
    Baris4 = 1
    Baris5 = 1
    If Kolom1 = 0 Then
        Keypad = 12
        Return
    End If
    If Kolom2 = 0 Then
        Keypad = 13
        Return
    End If
    If Kolom3 = 0 Then
        Keypad = 14
        Return
    End If
```

```
If Kolom4 = 0 Then
    Keypad = 15
    Return
End If
If Kolom5 = 0 Then
    Keypad = 16
    Return
End If
If Kolom6 = 0 Then
    Keypad = 17
    Return
End If
Baris1 = 1
Baris2 = 1
Baris3 = 1
Baris4 = 0
Baris5 = 1
If Kolom1 = 0 Then
    Keypad = 18
    Return
End If
If Kolom2 = 0 Then
    Keypad = 19
    Return
End If
If Kolom3 = 0 Then
    Keypad = 20
    Return
End If
If Kolom4 = 0 Then
    Keypad = 21
    Return
```

```
End If
If Kolom5 = 0 Then
    Keypad = 22
    Return
End If
If Kolom6 = 0 Then
    Keypad = 23
    Return
End If
Baris1 = 1
Baris2 = 1
Baris3 = 1
Baris4 = 1
Baris5 = 0
If Kolom2 = 0 Then
    Keypad = 24
    Return
End If
If Kolom3 = 0 Then
    Keypad = 26
    Return
End If
If Kolom4 = 0 Then
    Keypad = 27
    Return
End If
    If Kolom6 = 0 Then
        Keypad = 30
        Return
    End If
Return
```


Disply_huruf:

Select Case Tombol

Case 0 : Cls

Showpic 0 , 0 , A

SetFont Font8x8

Lcdat 4 , 70 , "HURUF A"

Case 1 : Cls

Showpic 0 , 0 , B

SetFont Font8x8

Lcdat 4 , 70 , "HURUF B"

Case 2 : Cls

Showpic 0 , 0 , C

SetFont Font8x8

Lcdat 4 , 70 , "HURUF C"

Case 3 : Cls

Showpic 0 , 0 , D

SetFont Font8x8

Lcdat 4 , 70 , "HURUF D"

Case 4 : Cls

Showpic 0 , 0 , E

SetFont Font8x8

Lcdat 4 , 70 , "HURUF E"

Case 5 : Cls

Showpic 0 , 0 , F

SetFont Font8x8

Lcdat 4 , 70 , "HURUF F"

Case 6 : Cls

Showpic 0 , 0 , G

SetFont Font8x8

Lcdat 4 , 70 , "HURUF G"

Case 7 : Cls

Showpic 0 , 0 , H

```

Setfont Font8x8
Lcdat 4 , 70 , "HURUF H"

```

Case 8 : Cls

```

Showpic 0 , 0 , I
Setfont Font8x8
Lcdat 4 , 70 , "HURUF I"

```

Case 9 : Cls

```

Showpic 0 , 0 , J
Setfont Font8x8
Lcdat 4 , 70 , "HURUF J"

```

Case 10 : Cls

```

Showpic 0 , 0 , K
Setfont Font8x8
Lcdat 4 , 70 , "HURUF K"

```

Case 11 : Cls

```

Showpic 0 , 0 , L
Setfont Font8x8
Lcdat 4 , 70 , "HURUF L"

```

Case 12 : Cls

```

Showpic 0 , 0 , M
Setfont Font8x8
Lcdat 4 , 70 , "HURUF M"

```

Case 13 : Cls

```

Showpic 0 , 0 , N
Setfont Font8x8
Lcdat 4 , 70 , "HURUF N"

```

Case 14 : Cls

```

Showpic 0 , 0 , O
Setfont Font8x8
Lcdat 4 , 70 , "HURUF O"

```

Case 15 : Cls

```

Showpic 0 , 0 , P

```

```
Setfont Font8x8
Lcdat 4 , 70 , "HURUF P"
```

Case 16 : CIs

```
Showpic 0 , 0 , Q
Setfont Font8x8
Lcdat 4 , 70 , "HURUF Q"
```

Case 17 : CIs

```
Showpic 0 , 0 , R
Setfont Font8x8
Lcdat 4 , 70 , "HURUF R"
```

Case 18 : CIs

```
Showpic 0 , 0 , S
Setfont Font8x8
Lcdat 4 , 70 , "HURUF S"
```

Case 19 : CIs

```
Showpic 0 , 0 , T
Setfont Font8x8
Lcdat 4 , 70 , "HURUF T"
```

Case 20 : CIs

```
Showpic 0 , 0 , U
Setfont Font8x8
Lcdat 4 , 70 , "HURUF U"
```

Case 21 : CIs

```
Showpic 0 , 0 , V
Setfont Font8x8
Lcdat 4 , 70 , "HURUF V"
```

Case 22 : CIs

```
Showpic 0 , 0 , W
Setfont Font8x8
Lcdat 4 , 70 , "HURUF W"
```

Case 23 : CIs

```
Showpic 0 , 0 , X
```

```
Setfont Font8x8
```

```
Lcdat 4 , 70 , "HURUF X"
```

```
Case 26 : Cls
```

```
Showpic 0 , 0 , Y
```

```
Setfont Font8x8
```

```
Lcdat 4 , 70 , "HURUF Y"
```

```
Case 27 : Cls
```

```
Showpic 0 , 0 , Z
```

```
Setfont Font8x8
```

```
Lcdat 4 , 70 , "HURUF Z"
```

```
Case 30 : Cls
```

```
Setfont Font 8x8
```

```
Lcdat 1 , 30 , "PETUNJUK"
```

```
Lcdat 3 , 1 , "Untuk memulai,"
```

```
Lcdat 4 , 1 , "tekan tombol"
```

```
Lcdat 5 , 1 , "A sampai z "
```

```
Lcdat 6 , 1 , "secara"
```

```
Lcdat 7 , 1 , "bergantian"
```

```
wait 4
```

```
Cls
```

```
Lcdat 1 , 30 , "PETUNJUK"
```

```
Lcdat 3 , 1 , "Gambar"
```

```
Lcdat 4 , 1 , "diperagakan"
```

```
Lcdat 5 , 1 , "dengan "
```

```
Lcdat 6 , 1 , "posisi tangan"
```

```
Lcdat 7 , 1 , "menghadap"
```

```
Lcdat 8 , 1 , "ke depan"
```

```
wait 4
```

```
Cls
```

```
Lcdat 1 , 30 , "PETUNJUK"
```

```
Lcdat 3 , 1 , "Gambar"
```

```
Lcdat 4 , 1 , "diperagakan"
```

```

Lcdat 5 , 1 , "dengan "
Lcdat 6 , 1 , "gerakan tangan"
Lcdat 7 , 1 , "menurut"
Lcdat 8 , 1 , "anak panah"
wait 4

```

```

Cls
Lcdat 1 , 30 , "PETUNJUK"
Lcdat 3 , 1 , "Untuk bantuan"
Lcdat 4 , 1 , "tekan tombol"
Lcdat 5 , 35 , "HELP"
wait 4

```

```

Cls
Lcdat 1 , 30 , "PETUNJUK"
Lcdat 3 , 1 , "Setelah selesai,"
Lcdat 4 , 1 , "matikan"
Lcdat 5 , 1 , "saklar ON/OFF"
Lcdat 6 , 1 , "dibagian bawah"
Lcdat 7 , 1 , "dan lepas"
Lcdat 8 , 1 , "kabel power"
wait 4

```

```

Cls
Lcdat 4 , 5 , "SELAMAT BELAJAR"
wait 1
End Select
Return
'end program

```

```

'Data file gambar

```

```

Uny:

```

```

$bgf "uny.bgf"

```

```

A:

```

```

$bgf "a.bgf"

```

B:
\$bgf "b.bgf"
C:
\$bgf "c.bgf"
D:
\$bgf "d.bgf"
E:
\$bgf "e.bgf"
F:
\$bgf "f.bgf"
G:
\$bgf "g.bgf"
H:
\$bgf "h.bgf"
I:
\$bgf "i.bgf"
J:
\$bgf "j.bgf"
K:
\$bgf "k.bgf"
L:
\$bgf "l.bgf"
M:
\$bgf "m.bgf"
N:
\$bgf "n.bgf"
O:
\$bgf "o.bgf"
P:
\$bgf "p.bgf"
Q:
\$bgf "q.bgf"

```
R:
$bgf "r.bgf"
S:
$bgf "s.bgf"
T:
$bgf "t.bgf"
U:
$bgf "u.bgf"
V:
$bgf "v.bgf"
W:
$bgf "w.bgf"
X:
$bgf "x.bgf"
Y:
$bgf "y.bgf"
Z:
$bgf "z.bgf"
$include "font8x8.font.bas"
```